Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

RANDOMIZED BLOCK COORDINATE DESCENDANT STRONG FOR LARGE-SCALE STOCHASTIC OPTIMIZATION

Wenyu Wang
Hong Wan

School of Industrial Engineering
Purdue University
West Lafayette, IN 47907-2023, USA

Kuo-Hao Chang

Department of Industrial Engineering
and Engineering Management
National Tsing Hua University
Hsinchu 30013, TAIWAN ROC

ABSTRACT

STRONG is a response surface methodology based algorithm that iteratively constructs linear or quadratic fitness model to guide the searching direction within the trust region. Despite its elegance and convergence, one bottleneck of the original STRONG in high-dimensional problems is the high cost per iteration. This paper proposes a new algorithm, RBC-STRONG, that extends the STRONG algorithm with the Random Coordinate Descent optimization framework. We proposed a RBC-STRONG algorithm and proved its convergence property. Our numerical experiments also show that RBC-STRONG achieves better computational performance than existing methods.

1 INTRODUCTION

Simulation models are widely used to analyze manufacturing, financial, computer, and service systems. One important function of these simulation models is to optimize the represented system through running simulations. We refer such optimization problems as Optimization via Simulation (OvS). Generally speaking, OvS is the problem of optimizing the expected performance of a stochastic system represented by a computer simulation model. For a recent comprehensive review of OvS, please refer to Fu et al. (2005) and Chau et al. (2014) .

The goal of this paper is to develop an efficient algorithm with cheap per-iteration cost to solve large-scale continuous OvS problem. The problem we consider in this paper is to optimize an unknown continuous function

$$\min g(\mathbf{x}) = \mathrm{E}[G(\mathbf{x}, \omega)]$$

where the underlying function $g(\mathbf{x})$ is defined by a black-box oracle. We assume that $g(\mathbf{x})$ can only be estimated through running simulation experiments at a particular value $\mathbf{x}$ and its gradient information is not available. The simulation output, $G(\mathbf{x}, \omega)$, is a function of the decision variable $\mathbf{x}$ and stochastic effect ω, and gives an unbiased estimator of the oracle function $g(\mathbf{x})$.

There are two main classes of methods for continuous OvS problem: direct and gradient method, and metamodel method (Barton 2009). Stochastic gradient (SG), such as the well-known Robbin-Monro method (Robbins and Monro 1951), and sample averaging approximation (SAA) methods (Verweij et al. 2003) are two of the best-known methods in the first class. The intuition of SG methods is the same as gradient descent. At each iteration, SG methods obtain an unbiased gradient estimation and search along this direction for a better solution. SG methods differ mainly in how to estimate the gradient. These methods typically have inexpensive iterations, but slow convergence and high sensitivity on algorithmic parameters, especially the step size (Kushner 2010). Moreover, SG methods may not converge in the general case. SAA methods, however, take an average of the functions and gradient estimators to reduce

the variance. SAA methods tend to be more robust with respect to parameters and converge much faster at a cost of more expensive iterations. For a comprehensive review on SG and SAA, please refers to Kim et al. (2015) and Kushner (2010) .

The second class, meta-model methods (Barton and Meckesheimer 2006, Barton 2009), however, does not use gradient directly. It is indirect because meta-model methods approximate oracle function $g(\mathbf{x})$ with surrogate functions $\hat{g}(\mathbf{x})$. With proper configuration, these surrogate functions are inexpensive to evaluate and easy to be updated and refined as the optimization progresses. Responses Surface Methodology (RSM) (Myers et al., 2009), Stochastic Kriging (Ankenman et al. 2009) , and MRAS (Hu et al., 2007) are all of this kind. Among these, RSM employs experimental design to build linear or quadratic local approximation to $g(\mathbf{x})$; stochastic Kriging interpolates oracle function $g(\mathbf{x})$ using Gaussian process; and MRAS takes the specific parameterized distribution family $f(\mathbf{x})$ as the reference distribution to approximate $g(\mathbf{x})$. Interested readers may refer to Myers et al. (2009), Ankenman et al. (2009), Hu et al. (2008) for in-depth introductions on these methods.

The new method proposed in this paper is a variant of STRONG (Stochastic Trust Region Response Surface Convergent Method) (Chang et al. 2011, Chang et al. 2012, Chang 2015) . It employs the idea of trust region method to search locally on a surrogate linear or quadratic model, which is approximated by response surface methodology. STRONG is guaranteed to converge to a stationary point in probability. Based on the STRONG framework, Chang (2015) proposed STRONG-X (X stands for relaxation) by relaxing the assumption of the normality of randomness and replacing hypothesis test by ratio test. Since STRONG-X is much more efficient and elegant compared with the original STRONG, in this paper, we build our method based upon STRONG-X, and, for brevity, refer STRONG-X as STRONG without distinguishing.

Notice that many, if not all, of these methods are intractable when the high dimension is an obstacle for either estimating gradient computation in direct and gradient method, or updating surrogate functions in metamodel methods. Recently, researchers proposed to apply *coordinate descent method* in direct and gradient methods as a remedy to solve various large-scale optimization problems (Nesterov 2012, Wang and Banerjee 2014). Coordinate descent method is among the first optimization methods studied in literature, but until recent study in large-scale problem it has not received much attention (Wright 2015).

For large-scale optimization problems, as working with all variables at each iteration may be costly, difficult or impossible, coordinate descent methods partition variables into manageable blocks and solve optimization problems by successively performing approximate minimization by focusing a single block only. This often drastically reduces the cost-per-iteration, making BCD methods simple and scalable. Both for their conceptual and algorithmic simplicity, BCD methods have been used in applications for many years, and their popularity continues to grow for their usefulness in training support vector machines in machine learning, Lasso, optimization, compressed image and so forth (Tseng and Yun 2007, Wang and Banerjee 2014, Wright 2015). At each iteration, coordinate descent method chooses one block of coordinates to sufficiently reduce the objective value while keeping all other blocks fixed. The main differences in all variants of coordinate descent methods consist in the criterion of choosing which coordinate to optimize at each iteration. Three most commonly used criteria in coordinate descent method are cyclic, random, and greedy descent coordinate search. Recent complexity result on these criteria were obtained by Xu and Yin (2015) . However, for cyclic coordinate descent, it is difficult to prove convergence, and almost impossible to estimate the rate of convergence. Greedy descent coordinate search assumes the knowledge of explicit form of gradient, which is conflict with our black-box assumptions. Thus, we consider the random coordinate descent method.

Note that Change et al. (2014) proposed STRONG-S to combine STRONG algorithm with factor screening procedure in order to solve large-scale OvS problems with the assumption on the sparsity of important variables. Chang et al. (2014) compared STRONG-S with the Simultaneous Perturbation Stochastic Approximation (SPSA) and the Modified NelderMead method (MNM) and concluded the

STRONG-S outperforms other methods in high-dimension scenarios. Thus, this paper uses STRONG-S as benchmark in numerical comparison.

In this paper, we develop a random block coordinate descent STRONG (RBC-STRONG) method suited for large-scale Stochastic Optimization problems. RBC-STRONG uses the coordinate descent method to separate the variable space into several blocks and iteratively optimize within each block. At each iteration, we randomly choose a block of variables and update this block using STRONG algorithm while fixing all other blocks of variables. Given mild conditions that the underlying function is Lipschitz continuous in each block, we show that RBC-STRONG converges to a stationary point in probability.

The outline of the paper is as follows. In Section 2, we formally state the research problem and its assumptions, review the STRONG algorithm and the randomized block coordinate descendant method, and then present the RBC-STRONG procedure. Section 3 gives the theoretical result on convergence rate. Then we study a numerical example and compare the performance of our proposed algorithm with STRONG-S in Section 4. We conclude with future research in Section 5.

2 THE RBC-STRONG ALGORITHM

2.1 Problem Definition

We consider a minimization problem with smooth but unknown objective function

$$\begin{aligned} &\min\ g(\mathbf{x}) = \mathbb{E}[G(\mathbf{x},\omega)] \\ &\text{s.t. } \mathbf{x} \in \mathscr{X} \end{aligned}$$

where $\mathbf{x}$ is the a $N \times 1$ vector within the feasible region $\mathscr{X}$; $g(\cdot) : \mathbf{x} \to \mathbb{R}^N$ is a continuous function defined by a black-box oracle; $G(\mathbf{x},\omega)$ is the stochastic response and is assumed to be measurable and perturbed by noise. Throughout the rest of the paper, we denote the scalar product by $\langle \mathbf{x},\mathbf{y}\rangle = \mathbf{x}^T\mathbf{y}$ and $\|\mathbf{x}\| = (\mathbf{x}^T\mathbf{x})^{\frac{1}{2}}$ for $\mathbf{x},\mathbf{y} \in \mathbb{R}$, and make the following assumptions on $g(\mathbf{x})$ and $G(\mathbf{x},\omega)$. Also, we use κ with acronyms for subscripts as notation for constant (Conn, Scheinberg, and Vicente 2009).

Assumption 1 The entire space $\mathbb{R}^N$ is decomposable into n blocks

$$N = \sum_{i=1}^{n} N_i$$

and vector $\mathbf{x}$ of decision variables has the partition:

$$\mathbf{x} = [x_1, ..., x_n]^T$$

Assumption 2 The objective function $g(\mathbf{x})$ is bounded below and has block-coordinate Lipschitz continuous gradient, i.e. there are Lipschitz constants $\kappa_{L_i} > 0$ such that:

$$\|\nabla_i g(\mathbf{x} + U_i h_i) - \nabla_i g(\mathbf{x})\| \le \kappa_{L_i} \|h_i\| \le \kappa_L \|h_i\| \ \ \forall \mathbf{x} \in \mathscr{X}, h_i \in \mathbb{R}^{N_i}, i = 1, ..., n$$

where $\kappa_L = \max_i \kappa_{L_i}$.

Assumption 3 There exists a positive constant κ_{bhm} such that for every k the Hessian matrix H_k of all estimations $\hat{g}(\mathbf{x})$ of $g(\mathbf{x})$ satisfy

$$\|H_k\| \le \kappa_{bhm} \tag{1}$$

Assumption 4 The fitted model $\hat{g}(\mathbf{x})$ from response surface methodology is a $(\kappa_{eg}, \kappa_{ef})$-fully linear model of $g(\mathbf{x})$ within the trust region Δ_T, such that for $\|\mathbf{s}\| \le \Delta_T$,

$$\|\nabla g(\mathbf{x}) - \nabla \hat{g}(\mathbf{x} + \mathbf{s})\| \le \kappa_{eg} \Delta_T \tag{2}$$

$$\|g(\mathbf{x}) - \hat{g}(\mathbf{x} + \mathbf{s})\| \le \kappa_{ef} \Delta_T^2 \tag{3}$$

Assumption 1 and 2 are used in random block coordinate descent. Readers can find similar variants in Tseng and Yun (2007), Patrascu and Necoara (2014), Xu and Yin (2015) . Under Assumption 1, $[x_1,...,x_n]^T$ are non-overlapping blocks of $\mathbf{x}$. Let $U_i \in \mathbb{R}^{N\times N_i}$ be the ith blocks of the identity matrix corresponding to ith block coordinates in $\mathbf{x}$,

$$I_n = [U_1,...,U_n]$$

Then we have $x_i = U_i^T\mathbf{x}$ and $\mathbf{x} = \sum_{i=1}^n U_i x_i$. In addition, the ith block in the gradient of the function g corresponding to x_i can be denoted as

$$\nabla_i g(\mathbf{x}) = U_i^T \nabla g(\mathbf{x})$$

Then, the model can be decomposed as

$$\min\ g(\mathbf{x}) = g(x_1,...,x_n) = \mathbb{E}[G(x_1,...,x_n,\omega)] \tag{4}$$

$$\text{s.t. } x_i \in \mathscr{X}_i,\ i = 1,2,...,n \tag{5}$$

where x_i is in the closed feasible region $\mathscr{X}_i \subseteq \mathbb{R}^{N_i}$ and $\mathscr{X} = \prod_{i=1}^n \mathscr{X}_i$.

Assumption 2 states that the partial gradients of $g(\mathbf{x})$ with respect to each block coordinate descent are Lipschitz continuous. Under the assumption 2 and the assumption that $g(\mathbf{x})$ is smooth, Patrascu and Necoara (2013) propose a variant of random block coordinate method and prove asymptotic convergence for the sequence to stationary points and sublinear rate of convergence. Note that Patrascu and Necoara (2013) assumes that knowledge of Lipschitz constants, L_i, which are unknown under black-box context.

Assumption 3 asserts a uniform bound on the model Hessian matrix. This assumption, however, is introduced for convenience. In the case of models that have large Hessian norms, we can simply replace the Hessian with matrix of a small norm (Banderia et al., 2014).

Assumption 4 ensures the response surface methodology approximates the oracle function with good accuracy. The definition of fully linear model is from Billups et al. (2013), Bandeira et al. (2014) . Notice that this assumption is stronger than assumption 3 in Chang (2015) , which assumes that the stochastic response $G(\mathbf{x},\omega)$ can be expressed as

$$G(\mathbf{x},\omega) = g(\mathbf{x}^k) + \left\langle \nabla g(\mathbf{x}^k), (\mathbf{x}-\mathbf{x}^k) \right\rangle + \frac{1}{2}(\mathbf{x}-\mathbf{x}^k)^T H(\mathbf{x}^k)(\mathbf{x}-\mathbf{x}^k) + \varepsilon_\omega$$

where ε_ω is a random noise with zero mean, $\mathbb{E}(\varepsilon_\omega) = 0$, and bounded variance, $Var(\varepsilon_\omega) = \sigma_\omega^2 < \infty$.

2.2 Review of STRONG

STRONG is a sequential procedure that iteratively employs experimental designs to fit a polynomial surrogate model in the trust region and uses the fitted model to find an optimal solution. Compared with the original trust region method, STRONG considers random responses. STRONG contains three stages. In stage I, STRONG approximates the underlying response surface with a first-order model. If the fitted model within the trust region provides a satisfactory new solution, then the algorithm continues the search with linear model; otherwise, the trust region will shrink. If the trust region becomes smaller than a user-specified threshold, then the algorithm will switch to stage II, where a second-order model will be used. If the second-order model fails to find a satisfactory solution, the algorithm will move to "Inner-Loop" and a series of second-order models will be constructed with accumulated design points.

In STRONG, each iteration applies response surface methodology. At each iteration, STRONG runs three steps: (i) fit a local model within the trust region, (ii) solve the following quadratic model, and (iii) apply ratio test and update trust region.

$$\min_{\mathbf{d}}\ \hat{g}(\mathbf{x}^k+\mathbf{d}) = \hat{g}(\mathbf{x}^k) + \left\langle \nabla\hat{g}(\mathbf{x}^k), \mathbf{d} \right\rangle + \frac{1}{2}\mathbf{d}^T\hat{H}_k\mathbf{d} \tag{6}$$

$$\text{s.t. } \|\mathbf{d}\| < \Delta_T \tag{7}$$

where $\mathbf{x}^k$ is the best solution after k iterations, $\nabla\hat{g}(\mathbf{x})$ and $\hat{H} = \nabla^2\hat{g}(\mathbf{x})$ are the estimated gradient and Hessian matrix, and Δ_T is the trust region.

Although the new solution is optimal for surrogate model, it may not be so for the oracle function. Thus, a ratio test is conducted to decide if the new solution is accepted or not. If the new solution is accepted, the trust region will expend or remain the same; otherwise the trust region will shrink. To control the error rate introduced by ratio test, STRONG requires the sample size of current and new solution N_c^{k+1} to increase at a certain rate, $N_c^{k+1} = \lceil N_c^k \cdot \psi \rceil$, where $\psi > 1$ (Chang 2015)

STRONG, however, is not for large-scale scenario since it is expensive, if still possible, to fit a linear or quadratic model when the dimension is high. Thus, we use Block Coordinate Descent method as a remedy to apply STRONG in large scale scenario.

2.3 Random Coordinate Descent for STRONG

To fit the problem in random coordinate descent framework, we decompose the variable space into n blocks. Instead of solving (6), we consider solving the problem

$$\min\ \hat{g}(\mathbf{x}) \tag{8}$$

$$\text{s.t. } \|x_i\| \leq \Delta_{Ti} \text{ for } i = 1,..,n \tag{9}$$

The trust region radius Δ_{T_i} are set in the following way: if the ith block of coordinates is chosen by RBC-STRONG, then let $\Delta_{T_i} = \Delta_T$ and $\Delta_{T_j} = 0$ for $j \neq i$. Thus, this formulation is equivalent to the STRONG optimization problem except that RBC-STRONG only optimizes one block of variables at one iteration. We can also formulate it as an optimization problem with regularization terms:

$$\min\ f(\mathbf{x}) := \hat{g}(\mathbf{x}) + \Omega(\mathbf{x})$$

where $\Omega(\mathbf{x})$ is the box indicator function

$$\Omega(\mathbf{x}) = \begin{cases} 0 & \text{if } \|x_i\| \leq \Delta_{Ti}\ \forall i = 1,...n \\ \infty & \text{otherwise.} \end{cases} \tag{10}$$

Notice that $\Omega(\mathbf{x})$ is block-separable, meaning $\Omega(\mathbf{x}) = \sum_{i=1}^n \Omega_i(x_i)$, and thus, for computing new solution for the ith block, we only need to consider the function $\Omega_i(x_i)$. The framework of random block coordinate is given in Algorithm 1.

An important property of our algorithm is that the direction d_{i_k} has the explicit expression since $\Omega(\mathbf{x})$ is a box constraint function:

$$d_{i_k} = \left[x_{i_k}^k - H_{i_k}^{-1}\nabla_{i_k} g(\mathbf{x}^k)\right]_{\Delta_T} \tag{11}$$

where $[x]_\Delta$ is the orthogonal projection of vector x on the trust region Δ.

Partial gradient $\nabla_i g(\mathbf{x})$ and partial Hessian matrix H_{i_k} are estimated through conducting an experimental design around $\mathbf{x}$. Notice that STRONG is a two-stage algorithm, where stage I fits a linear model and stage II fits a quadratic model. Corresponding, in our proposed RBC-STRONG, we only estimate the partial Hessian information H_{i_k} at stage II. For stage I, we set $H_i = 0$.

For probability distribution p_i, a general and robust choice of probability distribution p_i is uniform distribution, $p_i = \frac{1}{n}$ (Tseng and Mangasarian 2001, Wang and Banerjee 2014), where n is the number of blocks. We take this approach for both theoretical convergence proof and numerical evaluations. Moreover, Nesterov (2012) suggests choose random coordinate $i \in \{1,2,...,n\}$with probability $p_i = \frac{\kappa_{L_i}}{\sum_{j=1}^n \kappa_{L_j}}$ where $\kappa_{L_i}$is the coordinate Lipschitz constants. This approach, however, is unrealistic for black-box problems.

Algorithm 1 Random Block Coordinate Descent Procedure.

Inputs

Initial solution $\mathbf{x}^0 = (x_1^0, ..., x_n^0)$

Steps

procedure RCB($\mathbf{x}^0$)
  for $k = 0, 1, 2, ...$ **do**
    Randomly pick a block of coordinate i_k with probability p_{i_k}.
    Update $x_{i_k}^{k+1}$ with all other blocks fixed, $\mathbf{x}^{k+1} = \mathbf{x}^k + U_{i_k}^T d_{i_k}$, where the direction d_{i_k} is given by

$$d_{i_k} = \arg\min_{s_{i_k}} \left\langle \nabla_{i_k} g(\mathbf{x}^k), s_{i_k} \right\rangle + \frac{1}{2} s_{i_k}^T H_{i_k} s_{i_k} + \Omega_{\Delta_{i_k}}(s_{i_k})$$

  end for
  return $\mathbf{x}^k = (x_1^k, x_2^k, ..., x_n^k)$
end procedure

2.4 RBC-STRONG Algorithm

RBC-STRONG, same as STRONG, is a sequential procedure that employs both the trust region method and response surface methodology. The main difference is that RBC-STRONG applies random block coordinate descent method to reduce the per-iteration complexity. For each iteration, RBC-STRONG conducts four steps: (i) random pick block of coordinates i; (ii) use response surface methodology to fit a linear or quadratic model $\hat{g}(\mathbf{x})$; (iii) solve for a new candidate and evaluate it through simulation; and (iv) apply ratio test to the new candidate and update the trust region. The main structure appears in Figure 1, and the detailed algorithm show in Algorithm 2.

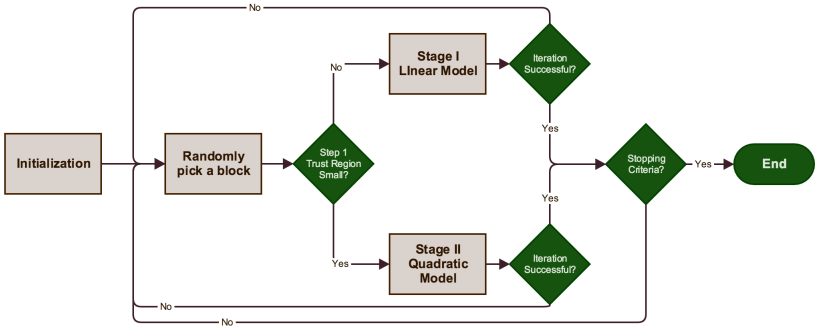

Figure 1: Main Framework of RBC-STRONG.

3 CONVERGENCE ANALYSIS

For general optimization problem, the first order necessary optimal condition is if $\mathbf{x}^\star \in \mathscr{X}$ is a local minimum, then $0 \in \nabla g(\mathbf{x}^\star)$. Any vector $\mathbf{x}^\star$ satisfying this relation is called a stationary point. Our theoretical analysis shows that under certain assumptions RBC-STRONG converges to a single stationary point by proving that $\lim_{k\to\infty} \left\|\nabla g(\mathbf{x}^k)\right\| = 0$ almost surely.

Before stating the theorem, we need an additional assumption that guarantees that the STRONG method are able to adequately minimize the model at each iteration of our algorithm.

Algorithm 2 RBC STRONG Main Algorithm.

Inputs

Initial solution $\mathbf{x}^0 = (x_1^0, ..., x_n^0)$;
Initial trust region radius Δ_0, maximum trust region radius $\Delta_{\max}$, switching trust region radius Δ_s;
Trust region parameters η_0, η_1, γ_1 and γ_2 ($0 < \eta_0 < \eta_1 < 1$ and $0 < \gamma_1 < 1 < \gamma_2$);
Sample size increasing rate ψ and initial central sample size n_c.

Steps

procedure RCBSTRONG($\mathbf{x}^0$, Δ_0, $\Delta_{\max}$, Δ_s, ψ, η_0, η_1, γ_1, γ_2)
  for $k = 0, 1, 2, ...$ **do**
    Random choose i_k with probability $p_{i_k} = \frac{1}{n}$;
    if $\Delta_{i_k} \leq \Delta_s$ **then**
      Fit a linear model $\hat{g}(\mathbf{x}) = \hat{g}(\mathbf{x}^k) + \langle \nabla \hat{g}(\mathbf{x}^k), \mathbf{d} \rangle$;
    else
      Fit a quadratic model $\hat{g}(\mathbf{x}) = \hat{g}(\mathbf{x}^k) + \langle \nabla \hat{g}(\mathbf{x}^k), \mathbf{d} \rangle + \frac{1}{2}\mathbf{d}^T \hat{H}_k \mathbf{d}$;
    end if
    Solve for new candidate $\mathbf{x}^{k+1} = \mathbf{x}^k + U_{i_k} x_{i_k}^k$ from 11;
    Simulate (evaluate) $G(\mathbf{x}^{k+1})$ for n_c times to get $G_{n_c}(\mathbf{x}^{k+1})$
    Compute reduction ratio $\rho^k = \frac{\text{observed achieved reduction}}{\text{predicted reduction}} = \frac{G_{n_c}(\mathbf{x}^{k+1}) - G_{n_c}(\mathbf{x}^k)}{\hat{g}(\mathbf{x}^{k+1}) - \hat{g}(\mathbf{x}^k)}$
    if $\rho^k < \eta_0$ **then**
      Reject $\mathbf{x}^{k+1}$, let $\mathbf{x}^{k+1} = \mathbf{x}^k$, $\Delta_{i_k} = \gamma_1 \Delta_{i_k}$
    else if $\eta_0 < \rho^k < \eta_1$ **then**
      Accept $\mathbf{x}^{k+1}$
    else
      Accept $\mathbf{x}^{k+1}$, let $\Delta_T^k = \min\{\gamma_2 \Delta_{i_k}\}$
    end if
    Update $n_c = \lceil \psi n_c \rceil$
  end for
  return $\mathbf{x}^k = (x_1^k, x_2^k, ..., x_n^k)$
end procedure

Assumption 5 For every k, and for all fitted model $\hat{g}(\mathbf{x}_k)$ from response surface methodology of oracle function $g(\mathbf{x}_k)$, we are able to compute a step $\mathbf{d}_k$ such that

$$\hat{g}(\mathbf{x}^k) - \hat{g}(\mathbf{x}^k + \mathbf{d}_k) \geq \frac{\kappa_{fcd}}{2} \left\| \nabla \hat{g}(\mathbf{x}^k) \right\| \min\{\hat{H}_k^{-1} \nabla \hat{g}(\mathbf{x}^k), \Delta_T\} \tag{12}$$

for some constant $\kappa_{fcd} \in (0, 1]$. We say in this case that $\mathbf{d}_k$ has achieved a fraction of Cauchy decrease.

This assumption is adapted from Conn et al. (2009), Billups et al. (2013), Shashaani et al. (2015). It is trivial to show that this inequality holds automatically for linear model since the Hessian matrix $H_k = 0$. For quadratic model, when $\kappa_{fcd} = 1$, the Cauchy step $\mathbf{d}_k$ minimizes the fitted model $\hat{g}(\mathbf{x})$ along the gradient direction $\nabla \hat{g}(\mathbf{x})$.

The first result demonstrates that the accumulated errors $\sum_{k=1}^{\infty} \left| \hat{g}^k(x^k) - g(x^k) \right|$ as well as the estimated accumulated reduction $\sum_{k=1}^{\infty} \left(\hat{g}^k(x^k) - \hat{g}^{k+1}(x^{k+1}) \right)$ is bounded above.

Corollary 1 Under assumption 1-5, for every realization of RBC-STRONG algorithm, with probability 1,

$$\sum_{k=1}^{\infty} \left| \hat{g}^k(x^k) - g(x^k) \right| < \infty \tag{13}$$

A direct observation from this corollary is that the estimated accumulated reduction $\sum_{k=1}^{\infty} \left(\hat{g}^k(x^k) - \hat{g}^{k+1}(x^{k+1}) \right)$ only differs from the true accumulated reduction $\sum_{k=1}^{\infty} (g(x^k) - g(x^{k+1}))$ by a constant.

Next two corollaries are crucial to the convergence of RBC-STRONG. Corollary 2 states that the trust region radius converges to zero as long as the Cauchy decrease is achieved (assumption 5). Then we shows in Corollary 3 that the fitted function gradient converges to the oracle function gradient almost surely, implying our main result that the oracle gradient converges to zero.

Corollary 2 Under assumption 1-5, for every realization of RBC-STRONG algorithm,

$$\Delta_{T_i} \xrightarrow{w.p.1} 0 \tag{14}$$

Corollary 3 Under assumption 1-5, for every realization of RBC-STRONG algorithm,

$$\left\| \nabla \hat{g}(\mathbf{x}^k) - \nabla g(\mathbf{x}^k) \right\| \xrightarrow{w.p.1} 0 \tag{15}$$

The main result of convergence is given by next theorem stating that the RBC-STRONG algorithm approaches to a stationary point of the oracle function almost surely.

Theorem 4 Under assumption 1-5, for every realization of RBC-STRONG algorithm,

$$\lim_{k \to \infty} \left\| \nabla g(\mathbf{x}^k) \right\| = 0$$

4 NUMERICAL EXPERIMENTS

In this section, we conduct two sets of numerical experiments: (i) compare RBC-STRONG with the STRONG-S method; (ii) test the performance of RBC-STRONG under different dimensions.

4.1 Test Problems

We use three functions as oracle function and add random noises to the oracle for simulation output.

The first test function is the extended **Rosenbrock function**

$$g(.) = \sum_{i=2}^{p} [100 \times (x_i - x_{i-1}^2)^2 + (x_{i-1} - 1)^2] \tag{16}$$

The Rosenbrock function is well known as a difficult minimization function. The extended Rosenbrock function is a multimodal function when the dimension $p \geq 4$. It achieves global minimum, 0, when $\mathbf{x} = [1, 1, ..., 1]$, and contains at least one local minimum near $\mathbf{x} = [-1, 1, ..., 1]$.

The second function is the **Freudenstein and Roth function**.

$$g(.) = \sum_{i=1}^{p/2} [-13 + x_{2i-1} + ((5 - x_{2i})x_{2i} - 2)x_{2i}]^2 + [-29 + x_{2i-1} + ((x_{2i} + 1)x_{2i} - 14)x_{2i}]^2 \tag{17}$$

Freudenstein and Roth function is also a multimodal function when $p \geq 2$. The global minimum value 0 is achieved at $\mathbf{x} = [5, 4, 5, 4, ..., 5, 4, ...]$.

The third function is the **Beala function**.

$$g(.) = \sum_{i}^{p/2} [1.5 - x_{2i-1}(1 - x_{2i})]^2 + [2.25 - x_{2i-1}(1 - x_{2i}^2)]^2 + [2.625 - x_{2i-1}(1 - x_{2i}^3)]^2 \tag{18}$$

Same as the aforementioned functions, the Beala function is multimodal. It is minimized at 0 when $\mathbf{x} = [3, 0.5, ...]$. Also, it has multiple local minimums.

We consider a variance structure $\varepsilon(\mathbf{x}) \sim N(0, a^2 \cdot g(\mathbf{x}))$ where $a = 0.1$ is a constant and $g(\mathbf{x})$ is the oracle value.

4.2 Algorithm Configuration

For fair comparison, we use the same STRONG configuration as Chang et al. (2014), except that we magnify the trust region size adjustment parameters, γ_1 and γ_2, because their scaling effects are weakened by decomposition in RBC-STRONG. Table 1 does not list all parameters for STRONG-S; interested readers may refer to Chang et al. (2014) for more information.

Table 1: Parameter Setting for STRONG.

Parameters	n_0	n_d	Δ_0	Δ_{switch}	$\Delta_{\max}$	η_0	η_1	γ_1	γ_2	block size
Settings	3	2	2	0.2	5	0.01	0.30	0.8	1.2	5

At each iteration, RBC-STRONG randomly select a (block of) coordinate from probability distribution p_i. Our convergence result is based on the uniform distribution, where $p_i = \frac{1}{n}$. For numerical evaluation, we proposed another distribution: $p_i \propto \Delta_i$, where Δ_i is the trust region radius of the ith block. The intuition is that the larger the radius, the more likely the function value reduces. We call these two choices of p_i as **uniform** and **weighted**, respectively.

4.3 Comparing with STRONG-S

This section is to compare RBC-STRONG with STRONG-S when the dimension $p = 200$. Chang et al. (2014) uses "optimality gap", *OG*, as performance criteria. This experiments, however, uses the optimality distance $g(\mathbf{x}^k) - g(\mathbf{x}^\star)$, in which $\mathbf{x}^\star$ is the true optima of the objective function and $\mathbf{x}^k$ is the solution returned by algorithms after k iterations. Notice that these two criteria are the same except that *OG* is normalized by the initial distance $g(\mathbf{x}^0) - g(\mathbf{x}^\star)$.

Table 2: Compare RBC-STRONG with STRONG-S.

Oracle function	**RBC-STRONG method**	**RBC-STRONG** $g(\mathbf{x}^k) - g(\mathbf{x}^\star)$	**STRONG-S** $g(\mathbf{x}^k) - g(\mathbf{x}^\star)$
Rosenbrock	Uniform	3084	69313
	Weight	2628	
Beala	Uniform	7208	140093
	Weight	6568	
Freudenstein and Roth	Uniform	26062	45676
	Weight	26210	

We run RBC-STRONG (uniform and weighted) and STRONG on each testing problem for 50 macro-replications with the fixed initial solution, $\mathbf{x}^0 = [20, 20, ..., 20]$. For each macro-replication, algorithm terminates after consuming $20,000$ function evaluations and returns the best solution been found $g(\mathbf{x}^k)$. The averages of these 50 macro-replications are reported in Table 2.

Notice that RBC-STRONG and STRONG-S have different assumptions on the underlying models. The table does not suggest that RBC-STRONG is superior and thus can replace STRONG-S, but rather to demonstrate the efficiency and robustness of RBC-STRONG for large-scale OvS problems.

4.4 Dealing with Dimensionality

RBC-STRONG is proposed to deal with large-scale OvS problems. It is, therefore, natural to ask how it performs as the dimension increases. To address this question, we use the same parameter configuration as in Table 1 and vary the dimension from 20 to 200. Table 3 reports the average function values as well as the optimization gap.

Table 3: Deal with Dimensionality.

Oracle function	**Dimension**	**Uniform**		**Weight**	
		$g(\mathbf{x}^k) - g(\mathbf{x}^\star)$	**OG**	$g(\mathbf{x}^k) - g(\mathbf{x}^\star)$	**OG**
Freudenstein and Roth	20	773.63	7.56302E-07	967.84	9.47741E-07
	40	2450.29	1.19897E-06	2431.50	1.18883E-06
	80	6903.15	1.68956E-06	7575.83	1.85088E-06
	120	10864.65	1.77164E-06	11852.46	1.93503E-06
	160	16193.38	1.98061E-06	17821.08	2.17754E-06
	200	26062.47	2.55016E-06	26210.17	2.56586E-06
Beala	20	63.00	2.45790E-10	97.82	3.81392E-10
	40	107.25	2.08936E-10	101.36	1.97240E-10
	80	411.47	4.00799E-10	472.75	4.60041E-10
	120	1751.90	1.13948E-09	1498.55	9.73092E-10
	160	2055.57	1.00064E-09	2596.58	1.26707E-09
	200	7208.70	2.80660E-09	6568.62	2.55922E-09
Rosenbrock	20	94.60	3.45167E-07	45.88	1.67147E-07
	40	345.02	6.12302E-07	139.76	2.47548E-07
	80	711.26	6.23565E-07	280.15	2.45528E-07
	120	1140.00	6.63406E-07	1032.06	6.00675E-07
	160	1753.63	7.63370E-07	1798.91	7.81348E-07
	200	3084.22	1.06981E-06	2628.33	9.22853E-07

A closer look at Table 3 reveals two interesting findings: (i) the optimization values $g(\mathbf{x}^k)$ are approximately proportional to dimensions p, (ii) the optimality gaps are roughly of the same order. Although convergence rate of coordinate descent is hard to prove, from this table, we speculate that RBC-STRONG approximately achieves convergence rate at $O(p/n)$, whereas RSM converges at the rate of $O(p^\kappa/n)$, where n is the simulation effort and $\kappa \approx 2$.

5 CONCLUSION AND FUTURE RESEARCH

In this paper, we proposed the RBC-STRONG algorithm to extend STRONG algorithm for the large-scale continuous variable black-box optimization problem. STRONG provides a powerful engine for OvS problems while it fails to handle large-scale problems due to the huge increase in per-iteration cost as the dimension hikes. Random coordinate descent, on the other hand, reduces the searching space dimension and thus forces STRONG algorithm to focus on small scale subproblem. Nevertheless, we substantially simplified the STRONG implementation by removing the inner loop in STRONG. So far, we proved the convergence properties of RBC-STRONG and demonstrated the efficiency in numerical evaluations. For future research, we identified four opening questions and research directions: **(1)** Convergence rate. Proving

convergence of BCD algorithms for minimization of non-convex objective functions is challenging (Richtrik and Tak 2012). We speculate the convergence rate of RBC-STRONG will be the same as STRONG in worst case scenario. **(2)** Regularization function. Trust region methods can be viewed as optimization with box-constraint function as regularization functions. In this perspective, other regularization forms, such as Lasso or Ridge regularization, may also be helpful (Wang and Banerjee 2014). **(3)** Shashaani et al. (2015) proposed an adaptive sampling schedule for trust region method. This new schedule is more economic and thus worth following. **(4)** Distributed algorithm. Due to the separability and randomization, the algorithm is adequate for distributed and parallel environments (Richtrik and Tak 2015).

REFERENCES

Ankenman, B., B. L. Nelson, and J. Staum. 2009, December. "Stochastic Kriging for Simulation Metamodeling". *Operations Research* 58 (2): 371–382.

Bandeira, A., K. Scheinberg, and L. Vicente. 2014, January. "Convergence of Trust-Region Methods Based on Probabilistic Models". *SIAM Journal on Optimization* 24 (3): 1238–1264.

Barton, R. R. 2009. "Simulation Optimization Using Metamodels". In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, 230–238. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Barton, R. R., and M. Meckesheimer. 2006. "Chapter 18 Metamodel-Based Simulation Optimization". In *Handbooks in Operations Research and Management Science*, edited by S. G. H. a. B. L. Nelson, Volume 13 of *Simulation*, 535–574. Elsevier.

Billups, S., J. Larson, and P. Graf. 2013. "Derivative-Free Optimization of Expensive Functions with Computational Error Using Weighted Regression". *SIAM Journal on Optimization* 23 (1): 27–53.

Chang, K.-H. 2015, May. "Improving the Efficiency and Efficacy of Stochastic Trust-Region Response-Surface Method for Simulation Optimization". *IEEE Transactions on Automatic Control* 60 (5): 1235–1243.

Chang, K.-H., L. J. Hong, and H. Wan. 2012. "Stochastic Trust-Region Response-Surface Method (STRONG) – A New Response-Surface Framework for Simulation Optimization". *INFORMS Journal on Computing* 25 (2): 230–243.

Chang, K.-H., M.-K. Li, and H. Wan. 2011. "Combining STRONG and Screening Designs for Large-Scale Simulation Optimization". In *Proceedings of the 2011 Winter Simulation Conference*, edited by S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, 4122–4133. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Chang, K.-H., M.-K. Li, and H. Wan. 2014. "Combining STRONG with Screening Designs for Large-Scale Simulation Optimization". *IIE Transactions* 46 (4): 357–373.

Chau, M., M. C. Fu, H. Qu, and I. O. Ryzhov. 2014. "Simulation Optimization: A Tutorial Overview and Recent Developments in Gradient-Based Methods". In *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S. D. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, 21–35. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Conn, A., K. Scheinberg, and L. Vicente. 2009. *Introduction to Derivative-Free Optimization*. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics.

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*, edited by 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.

Hu, J., M. C. Fu, and S. I. Marcus. 2007. "A Model Reference Adaptive Search Method for Global Optimization". *Operations Research* 55 (3): 549–568.

Hu, J., M. C. Fu, and S. I. Marcus. 2008. "A Model Reference Adaptive Search Method for Stochastic Global Optimization". *Communications in Information & Systems* 8 (3): 245–276.

Kim, S., R. Pasupathy, and S. G. Henderson. 2015. "A Guide to Sample Average Approximation". In *Handbook of Simulation Optimization*, edited by M. C. Fu, Number 216 in International Series in Operations Research & Management Science, 207–243. Springer New York.

Kushner, H. 2010. "Stochastic Approximation: A Survey". *Wiley Interdisciplinary Reviews: Computational Statistics* 2 (1): 87–96.

Myers, R. H., C. M. Anderson-Cook, and D. C. Montgomery. 2009. *Response Surface Methodology: Process and Product Optimization using Designed Experiments*. Hoboken, NJ: Wiley.

Necoara, I., and A. Patrascu. 2014, March. "A Random Coordinate Descent Algorithm for Optimization Problems with Composite Objective Function and Linear Coupled Constraints". *Computational Optimization and Applications* 57 (2): 307–337.

Nesterov, Y. 2012. "Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems". *SIAM Journal on Optimization* 22 (2): 341–362.

Patrascu, A., and I. Necoara. 2013. "Efficient Random Coordinate Descent Algorithms for Large-Scale Structured Nonconvex Optimization". *arXiv:1305.4027*.

Richtrik, P., and M. Tak. 2012, December. "Iteration Complexity of Randomized Block-Coordinate Descent Methods for Minimizing a Composite Function". *Mathematical Programming* 144 (1-2): 1–38.

Richtrik, P., and M. Tak. 2015. "Parallel Coordinate Descent Methods for Big Data Optimization". *Mathematical Programming* 156 (1-2): 433–484.

Robbins, H., and S. Monro. 1951. "A Stochastic Approximation Method". *The Annals of Mathematical Statistics* 22 (3): 400–407.

Shashaani, S., F. S. Hashemi, and R. Pasupathy. 2015. "ASTRO-DF: A Class of Adaptive Sampling Trust-Region Algorithms for Derivative-Free Simulation Optimization.". Under review.

Tseng, P., and C. O. L. Mangasarian. 2001. "Convergence of a Block Coordinate Descent Method for Nondifferentiable Minimization". *Journal of Optimazion Theory and Applications*:475–494.

Tseng, P., and S. Yun. 2007. "A Coordinate Gradient Descent Method for Nonsmooth Separable Minimization". *Mathematical Programming* 117 (1-2): 387–423.

Verweij, B., S. Ahmed, A. J. Kleywegt, G. Nemhauser, and A. Shapiro. 2003, February. "The Sample Average Approximation Method Applied to Stochastic Routing Problems: A Computational Study". *Computational Optimization and Applications* 24 (2-3): 289–333.

Wang, H., and A. Banerjee. 2014. "Randomized Block Coordinate Descent for Online and Stochastic Optimization". *arXiv:1407.0107*.

Wright, S. J. 2015, March. "Coordinate Descent Algorithms". *Mathematical Programming* 151 (1): 3–34.

Xu, Y., and W. Yin. 2015. "Block Stochastic Gradient Iteration for Convex and Nonconvex Optimization". *SIAM Journal on Optimization* 25 (3): 1686–1716.

AUTHOR BIOGRAPHIES

WENYU WANG is a Ph.D. student in the School of Industrial Engineering at Purdue University. His research interests include simulation optimization and stochastic modeling. His email address is wang1708@purdue.edu.

HONG WAN is Associate Professor in the School of Industrial Engineering at Purdue University. Her research interests include design and analysis of simulation experiments, simulation optimization, applied statistics, quality management, and healthcare engineering. She is a member of INFORMS and ASA. Her email address is hwan@purdue.edu and her web page is http://web.ics.purdue.edu/hwan.

KUO-HAO CHANG is an Associate Professor in Industrial Engineering and Engineering Management at National Tsing Hua University. His research interests include simulation optimization, stochastic models, and Monte Carlo simulation. His e-mail and website are chang@mx.nthu.edu.tw and https://sites.google.com/site/ssoptimizationlab/.