Stochastic Trust-Region Response-Surface Method (STRONG) – A New Response-Surface Framework for Simulation Optimization

Kuo-Hao Chang, L. Jeff Hong, Hong Wan

Zi Zhuang
School of Data Science
Fudan University

December 2021

Problem Definition

Consider the following simulation optimization problem:

$$\min_{x \in \mathbb{X}^p} \mathbb{E}[G(x)],$$

where

- x is the vector of continuous decision variables;
- $G(x)$ is the stochastic response evaluated at x;
- $\mathbb{X}^p$ is the p-dimensional decision space (or "region of operability" in terminology of the RSM)

RSM and trust region

RSM

- Constructs a first-order (linear) model, uses the model to find a better solution in a region of interest (ROI), and repeats the process;

RSM and trust region

RSM

- Constructs a first-order (linear) model, uses the model to find a better solution in a region of interest (ROI), and repeats the process;
- Once a first-order model is no longer appropriate, a second-order (quadratic) model is constructed to predict the optimal solution.

RSM and trust region

RSM

- Constructs a first-order (linear) model, uses the model to find a better solution in a region of interest (ROI), and repeats the process;
- Once a first-order model is no longer appropriate, a second-order (quadratic) model is constructed to predict the optimal solution.

Trust region

- A trust region at a solution x' with a radius $\Delta > 0$ is

$$B(x', \Delta) = \{x \in \mathbb{X}^p : \|x - x'\| \leq \Delta\}$$

Assumptions

1. The objective function $g(x)$ is bounded below, twice differentiable, and there exist two positive constants, α_1 and β_1, such that $\|\nabla g(x)\| \leq \alpha_1$ and $\|H(x)\| \leq \beta_1$.
2. The estimators of $g(x)$ and $\nabla g(x)$ satisfy $\sup_{x \in \mathbb{X}^p} |\bar{G}(x,n) - g(x)| \to 0 \ w.p.1$ as $n \to \infty$, and $\sup_{x \in \mathbb{X}^p} |\bar{D}(x,m) - \nabla g(x)| \to 0 \ w.p.1$ as $m \to \infty$.

First- and second-order models

First- and second-order model at iteration k:

$$r_k(x) = \hat{g}_k(x_k) + \hat{\nabla} g_k^T(x_k)(x - x_k),$$

$$\mathsf{r}_k(x) = \hat{g}_k(x_k) + \hat{\nabla} g_k^T(x_k)(x - x_k) + \frac{1}{2}(x - x_k)^T \hat{H}_k(x_k)(x - x_k).$$

First- and second-order models

First- and second-order model at iteration k:

$$r_k(x) = \hat{g}_k(x_k) + \hat{\nabla} g_k^T(x_k)(x - x_k),$$

$$\mathrm{r}_k(x) = \hat{g}_k(x_k) + \hat{\nabla} g_k^T(x_k)(x - x_k) + \frac{1}{2}(x - x_k)^T \hat{H}_k(x_k)(x - x_k).$$

Solution (ideally):

$$x_k^* \in argmin\{r_k(x) : x \in B(x_k, \Delta_k)\},$$

First- and second-order models

First- and second-order model at iteration k:

$$r_k(x) = \hat{g}_k(x_k) + \hat{\nabla} g_k^T(x_k)(x - x_k),$$

$$\mathrm{r}_k(x) = \hat{g}_k(x_k) + \hat{\nabla} g_k^T(x_k)(x - x_k) + \frac{1}{2}(x - x_k)^T \hat{H}_k(x_k)(x - x_k).$$

Solution (ideally):

$$x_k^* \in argmin\{r_k(x) : x \in B(x_k, \Delta_k)\},$$

Cauchy Point Calculation:

1. Find the steepest descent direction $d_k = argmin\{\hat{g}_k(x_k) + \hat{\nabla} g_k^T(x_k) d : \|d\| \leq \Delta_k\}$;
2. Choose a step size $\tau_k = argmin\{r_k(\tau d_k) : \tau > 0, \|\tau d_k\| \leq \Delta_k\}$.
3. Let $x_k^* = x_k + \tau_k d_k$.

RC (Ratio-Comparison) test

The RC test computes

$$\rho_k = \frac{\hat{g}_k(x_k) - \hat{g}_k(x_k^*)}{r_k(x_k) - r_k(x_k^*)}$$

RC (Ratio-Comparison) test

The RC test computes

$$\rho_k = \frac{\hat{g}_k(x_k) - \hat{g}_k(x_k^*)}{r_k(x_k) - r_k(x_k^*)}$$

Let $0 < \eta_0 < \eta_1 < 1$, for example, set $\eta_0 = 1/4, \eta_1 = 3/4$.

- If ρ_k is large ($\rho_k \geq \eta_1$), which implies that the new observed solution is significantly better than the current one, the new solution will be accepted.

RC (Ratio-Comparison) test

The RC test computes

$$\rho_k = \frac{\hat{g}_k(x_k) - \hat{g}_k(x_k^*)}{r_k(x_k) - r_k(x_k^*)}$$

Let $0 < \eta_0 < \eta_1 < 1$, for example, set $\eta_0 = 1/4, \eta_1 = 3/4$.

- If ρ_k is large ($\rho_k \geq \eta_1$), which implies that the new observed solution is significantly better than the current one, the new solution will be accepted.
- If ρ_k is moderate ($\eta_0 \leq \rho_k < \eta_1$), which implies that the "observed reduction" has fair agreements with the "predicted reduction", the procedure will also accept the new solution.

RC (Ratio-Comparison) test

The RC test computes

$$\rho_k = \frac{\hat{g}_k(x_k) - \hat{g}_k(x_k^*)}{r_k(x_k) - r_k(x_k^*)}$$

Let $0 < \eta_0 < \eta_1 < 1$, for example, set $\eta_0 = 1/4, \eta_1 = 3/4$.

- If ρ_k is large ($\rho_k \geq \eta_1$), which implies that the new observed solution is significantly better than the current one, the new solution will be accepted.
- If ρ_k is moderate ($\eta_0 \leq \rho_k < \eta_1$), which implies that the "observed reduction" has fair agreements with the "predicted reduction", the procedure will also accept the new solution.
- If ρ_k is close to 0 or negative ($\rho_k < \eta_0$), which implies that the "observed reduction" does not agree with the "predicted reduction", the new solution will be rejected.

SR (Sufficient-Reduction) test

The SR test is defined as

$$H_0 : g(x_k) - g(x_k^*) \leq \eta_0^2 \zeta_k \quad v.s. \quad H_1 : g(x_k) - g(x_k^*) > \eta_0^2 \zeta_k,$$

Let $0 < \eta_0 < \eta_1 < 1$, for example, set $\eta_0 = 1/4, \eta_1 = 3/4.$ where

$$\begin{aligned} \zeta_k &:= \|\hat{\nabla} g_k(x_k)\| \Delta_k, \\ \zeta_k &:= \frac{1}{2} \|\hat{\nabla} g_k(x_k)\| \min\{\frac{\|\hat{\nabla} g_k(x_k)\|}{\hat{\|} H_k(x_k)\|}, \Delta_k\}. \end{aligned}$$

in first- or second-order models respectively. If H_0 is rejected, we then conclude that the new solution yields a sufficient reduction.

The STRONG Algorithm

STRONG has two stages and an inner loop.

The STRONG Algorithm

STRONG has two stages and an inner loop.
At iteration k, STRONG conducts the following four steps:

1. Construct a local model $r_k(x)$ around the center point x_k;

The STRONG Algorithm

STRONG has two stages and an inner loop.
At iteration k, STRONG conducts the following four steps:

1. Construct a local model $r_k(x)$ around the center point x_k;
2. Solve $x_k^* \in argmin\{r_k(x) : x \in B(x_k, \Delta_k)\}$;

The STRONG Algorithm

STRONG has two stages and an inner loop.
At iteration k, STRONG conducts the following four steps:

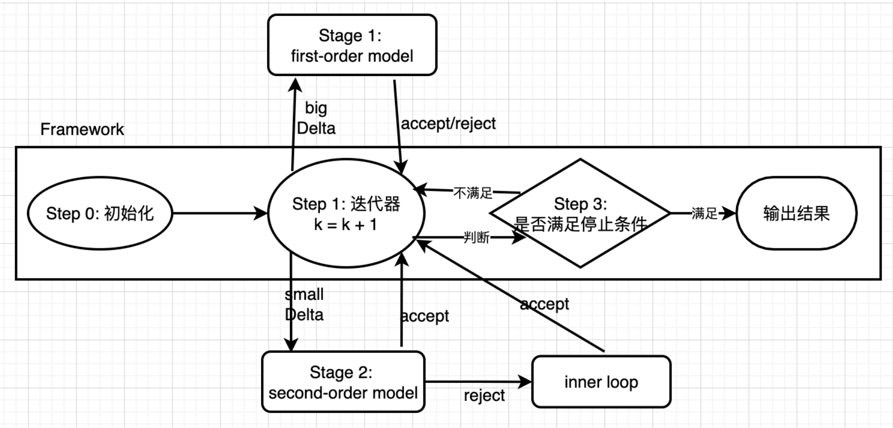
1. Construct a local model $r_k(x)$ around the center point x_k;
2. Solve $x_k^* \in argmin\{r_k(x) : x \in B(x_k, \Delta_k)\}$;
3. Simulate several observations at x_k^* and estimate $g(x_k^*)$;

The STRONG Algorithm

STRONG has two stages and an inner loop.
At iteration k, STRONG conducts the following four steps:

1. Construct a local model $r_k(x)$ around the center point x_k;
2. Solve $x_k^* \in argmin\{r_k(x) : x \in B(x_k, \Delta_k)\}$;
3. Simulate several observations at x_k^* and estimate $g(x_k^*)$;
4. Conduct ratio-comparison (RC) and sufficient-reduction (SR) tests to examine the quality of x^k and to update x_{k+1} and the size of trust region Δ_{k+1}.

The STRONG Algorithm



STRONG-Framework

- Step 0. Set the iteration count $k = 0$. Select an initial solution x_0, initial sample size of the center point n_0 and of the gradient estimator m_0, initial trust region size Δ_0, the switch threshold $\tilde{\Delta}$ satisfying $\Delta_0 > \tilde{\Delta} > 0$, and constants η_0, η_1, γ_1, and γ_2 satisfying $0 < \eta_0 < \eta_1 < 1$ and $0 < \gamma_1 < 1 < \gamma_2$.

STRONG-Framework

- Step 0. Set the iteration count $k = 0$. Select an initial solution x_0, initial sample size of the center point n_0 and of the gradient estimator m_0, initial trust region size Δ_0, the switch threshold $\tilde{\Delta}$ satisfying $\Delta_0 > \tilde{\Delta} > 0$, and constants η_0, η_1, γ_1, and γ_2 satisfying $0 < \eta_0 < \eta_1 < 1$ and $0 < \gamma_1 < 1 < \gamma_2$.
- Step 1. Let $k = k + 1$. If $\Delta_k > \tilde{\Delta}$, go to Stage I; otherwise, go to Stage II.

STRONG-Framework

- Step 0. Set the iteration count $k = 0$. Select an initial solution x_0, initial sample size of the center point n_0 and of the gradient estimator m_0, initial trust region size Δ_0, the switch threshold $\tilde{\Delta}$ satisfying $\Delta_0 > \tilde{\Delta} > 0$, and constants η_0, η_1, γ_1, and γ_2 satisfying $0 < \eta_0 < \eta_1 < 1$ and $0 < \gamma_1 < 1 < \gamma_2$.
- Step 1. Let $k = k + 1$. If $\Delta_k > \tilde{\Delta}$, go to Stage I; otherwise, go to Stage II.
- Step 2. If the termination criterion is satisfied, stop and return the solution. Otherwise, go to Step 1.

STRONG-Stage I

1. Build a first-order model.

STRONG-Stage I

1. Build a first-order model.
2. Solve the subproblem and obtain a new solution x_k^*.

STRONG-Stage I

1. Build a first-order model.
2. Solve the subproblem and obtain a new solution x_k^*.
3. Take n_0 replications at the new solution x_k^*.

STRONG-Stage I

1. Build a first-order model.
2. Solve the subproblem and obtain a new solution x_k^*.
3. Take n_0 replications at the new solution x_k^*.
4. Perform the RC test, update the size of trust region:

STRONG-Stage I

1. Build a first-order model.
2. Solve the subproblem and obtain a new solution x_k^*.
3. Take n_0 replications at the new solution x_k^*.
4. Perform the RC test, update the size of trust region:
   - (Reject) let $x_{k+1} = x_k$, $\Delta_{k+1} = \gamma_1 \Delta_k$ and return to Step 1 in the main framework.

STRONG-Stage I

1. Build a first-order model.
2. Solve the subproblem and obtain a new solution x_k^*.
3. Take n_0 replications at the new solution x_k^*.
4. Perform the RC test, update the size of trust region:
   - (Reject) let $x_{k+1} = x_k$, $\Delta_{k+1} = \gamma_1 \Delta_k$ and return to Step 1 in the main framework.
   - (Moderate accept) perform the SR test:

STRONG-Stage I

1. Build a first-order model.
2. Solve the subproblem and obtain a new solution x_k^*.
3. Take n_0 replications at the new solution x_k^*.
4. Perform the RC test, update the size of trust region:
   - (Reject) let $x_{k+1} = x_k$, $\Delta_{k+1} = \gamma_1 \Delta_k$ and return to Step 1 in the main framework.
   - (Moderate accept) perform the SR test:
     - If passed, let $x_{k+1} = x_k^*$ and $\Delta_{k+1} = \Delta_k$, and return to Step 1 in the main framework.

STRONG-Stage I

1. Build a first-order model.
2. Solve the subproblem and obtain a new solution x_k^*.
3. Take n_0 replications at the new solution x_k^*.
4. Perform the RC test, update the size of trust region:
   - (Reject) let $x_{k+1} = x_k$, $\Delta_{k+1} = \gamma_1 \Delta_k$ and return to Step 1 in the main framework.
   - (Moderate accept) perform the SR test:
     - If passed, let $x_{k+1} = x_k^*$ and $\Delta_{k+1} = \Delta_k$, and return to Step 1 in the main framework.
     - Otherwise, let $x_{k+1} = x_k$, $\Delta_{k+1} = \gamma_1 \Delta_k$ and return to Step 1 in the main framework.

STRONG-Stage I

1. Build a first-order model.
2. Solve the subproblem and obtain a new solution x_k^*.
3. Take n_0 replications at the new solution x_k^*.
4. Perform the RC test, update the size of trust region:
   - (Reject) let $x_{k+1} = x_k$, $\Delta_{k+1} = \gamma_1 \Delta_k$ and return to Step 1 in the main framework.
   - (Moderate accept) perform the SR test:
     - If passed, let $x_{k+1} = x_k^*$ and $\Delta_{k+1} = \Delta_k$, and return to Step 1 in the main framework.
     - Otherwise, let $x_{k+1} = x_k$, $\Delta_{k+1} = \gamma_1 \Delta_k$ and return to Step 1 in the main framework.
   - (Fully accept) perform the SR test:

STRONG-Stage I

1. Build a first-order model.
2. Solve the subproblem and obtain a new solution x_k^*.
3. Take n_0 replications at the new solution x_k^*.
4. Perform the RC test, update the size of trust region:
   - (Reject) let $x_{k+1} = x_k$, $\Delta_{k+1} = \gamma_1 \Delta_k$ and return to Step 1 in the main framework.
   - (Moderate accept) perform the SR test:
     - If passed, let $x_{k+1} = x_k^*$ and $\Delta_{k+1} = \Delta_k$, and return to Step 1 in the main framework.
     - Otherwise, let $x_{k+1} = x_k$, $\Delta_{k+1} = \gamma_1 \Delta_k$ and return to Step 1 in the main framework.
   - (Fully accept) perform the SR test:
     - If passed, let $x_{k+1} = x_k^*$ and $\Delta_{k+1} = \gamma_2 \Delta_k$, and return to Step 1 in the main framework.

STRONG-Stage I

1. Build a first-order model.
2. Solve the subproblem and obtain a new solution x_k^*.
3. Take n_0 replications at the new solution x_k^*.
4. Perform the RC test, update the size of trust region:
   - (Reject) let $x_{k+1} = x_k$, $\Delta_{k+1} = \gamma_1 \Delta_k$ and return to Step 1 in the main framework.
   - (Moderate accept) perform the SR test:
     - If passed, let $x_{k+1} = x_k^*$ and $\Delta_{k+1} = \Delta_k$, and return to Step 1 in the main framework.
     - Otherwise, let $x_{k+1} = x_k$, $\Delta_{k+1} = \gamma_1 \Delta_k$ and return to Step 1 in the main framework.
   - (Fully accept) perform the SR test:
     - If passed, let $x_{k+1} = x_k^*$ and $\Delta_{k+1} = \gamma_2 \Delta_k$, and return to Step 1 in the main framework.
     - Otherwise, let $x_{k+1} = x_k$, $\Delta_{k+1} = \gamma_1 \Delta_k$ and return to Step 1 in the main framework.

STRONG-Stage II

1. Build a second-order model.
2. ......
3. ......
4. Perform the RC test, update the size of trust region:

STRONG-Stage II

1. Build a second-order model.
2. ......
3. ......
4. Perform the RC test, update the size of trust region:
   - (Reject) ......, go to inner loop.

STRONG-Stage II

1. Build a second-order model.
2. ......
3. ......
4. Perform the RC test, update the size of trust region:
   - (Reject) ......, go to inner loop.
   - (Moderate accept) perform the SR test:
     - If passed, ..., return to Step 1 in the main framework.
     - Otherwise, go to inner loop.

STRONG-Stage II

1. Build a second-order model.
2. ......
3. ......
4. Perform the RC test, update the size of trust region:
   - (Reject) ......, go to inner loop.
   - (Moderate accept) perform the SR test:
     - If passed, ..., return to Step 1 in the main framework.
     - Otherwise, go to inner loop.
   - (Fully accept) perform the SR test:
     - If passed,......, return to Step 1 in the main framework.
     - Otherwise, go to inner loop.

STRONG-Inner loop

Inner loop:

- Stages I and II both may find no satisfactory solution that can pass the RC and SR tests.

STRONG-Inner loop

Inner loop:

- Stages I and II both may find no satisfactory solution that can pass the RC and SR tests.
- The inner loop is designed so that it can always find a satisfactory solution.

STRONG-Inner loop

Inner loop:

- Stages I and II both may find no satisfactory solution that can pass the RC and SR tests.
- The inner loop is designed so that it can always find a satisfactory solution.

Process:

1. Set subiteration count $i = 0$. Let $n_{k_0} = n_0$, $m_{k_0} = m_0$, and $\Delta k_0 = \Delta k$.

STRONG-Inner loop

Inner loop:

- Stages I and II both may find no satisfactory solution that can pass the RC and SR tests.
- The inner loop is designed so that it can always find a satisfactory solution.

Process:

1. Set subiteration count $i = 0$. Let $n_{k_0} = n_0$, $m_{k_0} = m_0$, and $\Delta k_0 = \Delta k$.
2. Let $i = i + 1$. Increase the sample sizes for the current and new solutions, satisfies

$$n^*_{k_i} \geq (1/\gamma_1^4 + 1) n^*_{k_{i-1}},$$
$$n_{k_i} = \max\{n_{k_{i-1}}, n^*_{k_i}\}.$$

STRONG-Inner loop

Inner loop:

- Stages I and II both may find no satisfactory solution that can pass the RC and SR tests.
- The inner loop is designed so that it can always find a satisfactory solution.

Process:

1. Set subiteration count $i = 0$. Let $n_{k_0} = n_0$, $m_{k_0} = m_0$, and $\Delta k_0 = \Delta k$.
2. Let $i = i + 1$. Increase the sample sizes for the current and new solutions, satisfies

$$n^*_{k_i} \geq (1/\gamma_1^4 + 1) n^*_{k_{i-1}},$$
$$n_{k_i} = \max\{n_{k_{i-1}}, n^*_{k_i}\}.$$

3. Increase the sample size of the gradient estimator, satisfies

$$m_{k_i} \geq (1/\gamma_1^2 + 1) m_{k_{i-1}}.$$

STRONG-Inner loop

4. Build a second-order model.
5. Solve the subproblem and obtain a new solution $x^*_{k_i}$.
6. Perform the RC test, update the size of trust region:

STRONG-Inner loop

4. Build a second-order model.
5. Solve the subproblem and obtain a new solution $x^*_{k_i}$.
6. Perform the RC test, update the size of trust region:
   - (Reject), let $\Delta_{k_{i+1}} = \gamma_1 \Delta_{k_i}$, go back to Step 2 in inner loop.

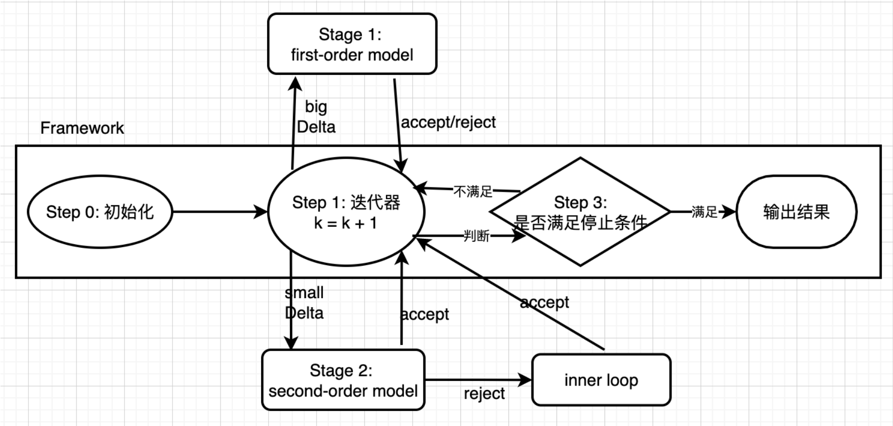
STRONG-Inner loop

4. Build a second-order model.
5. Solve the subproblem and obtain a new solution $x^*_{k_i}$.
6. Perform the RC test, update the size of trust region:
   - (Reject), let $\Delta_{k_{i+1}} = \gamma_1 \Delta_{k_i}$, go back to Step 2 in inner loop.
   - (Moderate accept), perform the SR test:
     - If passed, ..., return to Step 1 in the main framework.
     - Otherwise, go back to Step 2 in inner loop.

STRONG-Inner loop

4. Build a second-order model.
5. Solve the subproblem and obtain a new solution $x^*_{k_i}$.
6. Perform the RC test, update the size of trust region:
   - (Reject), let $\Delta_{k_{i+1}} = \gamma_1 \Delta_{k_i}$, go back to Step 2 in inner loop.
   - (Moderate accept), perform the SR test:
     - If passed, ..., return to Step 1 in the main framework.
     - Otherwise, go back to Step 2 in inner loop.
   - (Fully accept), perform the SR test:
     - If passed, ..., return to Step 1 in the main framework.
     - Otherwise, go back to Step 2 in inner loop.

The STRONG Algorithm



Thanks

Thanks for listening.