See arXiv link **HERE**

SPSA-Based Switch Updating Algorithm for Constrained Stochastic Optimization [*]

Zhichao Jia Ziyi Wei James C. Spall

Abstract

Simultaneous perturbation stochastic approximation (SPSA) is widely used in stochastic optimization due to its high efficiency, asymptotic stability, and reduced number of required loss function measurements. However, the standard SPSA algorithm needs to be modified to deal with constrained problems. In recent years, sequential quadratic programming (SQP)-based projection ideas and penalty ideas have been analyzed. Both ideas have convergence results and a potentially wide range of applications, but with some limitations in practical consideration, such as computation time complexity and feasibility guarantee. We propose an SPSA-based switch updating algorithm, which alternatively updates based on loss function or one of the inequality constraints, depending on current feasibility in each iteration. We show convergence and asymptotic results on the number of loss function measurements for this algorithm, and analyze its properties relative to other methods. We numerically compare the switch updating algorithm with the penalty function approach for two constrained problems.

1 Introduction

Stochastic approximation (SA) is a family of stochastic algorithms finding the extremums of loss functions with only noisy measurements of them being available. To solve unconstrained stochastic optimization problems, several SA methods have been proposed. The simultaneous perturbation stochastic approximation (SPSA) algorithm analyzed in [1] is an efficient first-order stochastic approximation that requires only two noisy loss function measurements in each iteration. Applications of the SPSA algorithm can be found in [2–4], and many other sources. However, the classic SPSA algorithm needs modifications when it needs to fit constrained problem cases. This paper proposes a simple but effective and feasibility-guaranteed algorithm based on the SPSA algorithm to deal with some kinds of inequality constraints. The algorithm relies on the ideas of switching updating rules according to different feasibility conditions.

Let us now describe our problem setting. We consider the following constrained optimization problem:

$$\begin{aligned} &\min L(\theta), \\ &\text{s.t. } q_i(\theta) \leq 0, \qquad i = 1, ..., m, \end{aligned} \tag{1.1}$$

where $L : \mathbb{R}^p \to \mathbb{R}$ is the objective loss function, and $q_i : \mathbb{R}^p \to \mathbb{R}, i = 1, ..., m$ are differentiable constraint functions. For $L(\theta)$, only noisy function measurements $y(\theta) = L(\theta) + \varepsilon(\theta)$ are available, where $\varepsilon(\theta)$ represents the noise term. To ensure a p-dimensional feasible region $\Theta = \{\theta | q_i(\theta) \leq 0, i = 1, ..., m\}$, we assume that the interior of Θ is not an empty set. The solution θ^* is assumed to be unique and may lie in anywhere in Θ (interior or boundary).

[*] The is a drafted version of our current work, to be updated in future. We are also organizing our scratch work dealing with the noisy constraints. We will be able to show this part of work once we finish our polishing.

Many technical approaches in [5] have proved useful in constrained problem settings. Projection to the feasible set Θ after updating in each iteration is a common way to maintain feasibility, and, in some cases, it can be simply realized in the SPSA algorithm. For example, [6] and [7] applied the idea of projecting current infeasible points back to their nearest points in Θ. However, constraint regions are often sufficiently complex so that no way exists to locate exact projection points during iterations, making the projection idea impractical. In recent years, [8] applied a sequential quadratic programming (SQP)-based projection method to the SPSA algorithm, searching for approximate projection points by solving an equivalently converted deterministic quadratic constrained problem with the SQP algorithm in each iteration. With approximate projection points converging to exact projections during iterations, the modified projection SPSA algorithm shows convergence as well. This makes the projection idea useful when facing complex constraints. However, when the exact solution lies on the boundary of Θ, solving a constrained problem using the projection process in each of the possibly huge number of iterations will consume much computational time.

Another frequently utilized notion is to introduce penalty functions to stochastic approximation (SA) algorithms (see e.g. [9], Chap. 5 and Chap. 6). By properly constructing a penalty term related to all constraints and adding it to the loss function, the constrained problem is converted to an equivalent unconstrained problem, which eliminates the need to explicitly consider or maintain feasibility during iterations and greatly reduces the computational complexity for each iteration. For instance, [10] constructed their modified loss functions with several different kinds of penalty terms and solved these unconstrained problems by the SPSA algorithm. The penalty function approach is very sensitive to the choice of weighting for the penalty. Well chosen hyper-parameter values associated with the penalty provide better performance, while poor choices impair the convergence of the algorithm. Further, the penalty-based method generally fails to guarantee feasibility for any practical solution from a finite number of iterations. Generally, although implementation of this idea avoids focusing on feasibility in each iteration, the challenges in implementing penalty functions are significant.

In this paper, we consider advantages and drawbacks of the SPSA algorithms applied with projection ideas and penalty ideas, and propose an algorithm based on the SPSA algorithm to cope with constrained problems of the form in (1.1) under some different assumptions. With up to m feasibility evaluations on the current point in each iteration, our algorithm chooses either an SPSA step based on two measurements of the objective loss function $L(\theta)$ or a gradient descent step based on one of the infeasible explicit constraint functions $q_i(\theta)$. This results in an updating formula changes to θ guided by the loss function and changes to θ guided by the constraints. Therefore, this SPSA-based switch updating (SU) algorithm could directly compute and run each step during iterations, eliminating extra hyper-parameter selection without dealing with any time-consuming process to attain feasibility, such as making projections to the feasible set Θ. Further, the algorithm ensures feasibility for the final solution. This appears to be a novel improvement of SPSA in the constrained stochastic optimization setting.

The organization of the rest of the paper is as follows. Section 2 proposes the SPSA-based switch updating (SU) algorithm. Section 3 studies its convergence. Section 4 gives an upper bound of its convergence rate. Section 5 presents numerical experiments, comparing the performance of the SU algorithm with the SPSA algorithms using different kinds of penalty ideas. Finally, Section 6 offers closing remarks.

2 SPSA-based Switch Updating Algorithm

The switch updating method solves the deterministic problem as the form of (1.1) by executing the step as below:

$$\theta_{t+1} = \theta_t - a_t h_t, \tag{2.1}$$

where

$$h_t = \begin{cases} \nabla q_1(\theta_t) & q_1(\theta_t) > 0, \\ \nabla q_2(\theta_t) & q_1(\theta_t) \leq 0, q_2(\theta_t) > 0, \\ \nabla q_3(\theta_t) & q_1(\theta_t), q_2(\theta_t) \leq 0, q_3(\theta_t) > 0, \\ \cdots\cdots & \\ \nabla L(\theta_t) & q_i(\theta_t) \leq 0, i = 1, 2, ..., m. \end{cases} \tag{2.2}$$

The basic idea for the switch updating method is that, as long as θ_t is infeasible, we choose the gradient of one of the infeasible constraints to pull the point back to the feasible region, and then perform the deterministic first-order gradient descent steps. Such a switch updating idea was proposed in [11] dealing with only one constraint function (i.e. $m = 1$). This sheds light on solving the constrained stochastic optimization problems.

The SPSA algorithm analyzed in [12], Chap.7 is based on the use of noisy measurements of $L(\theta)$, say $y(\theta) = L(\theta) + \varepsilon(\theta)$, to approximate the gradient of L of any valid θ without the need of $L(\theta)$ and its gradient $g(\theta)$. It only uses two measurements $y(\theta)$ to update the estimate of θ via an efficient gradient estimate in each step. To combine the classic SPSA algorithm with the idea of the switch updating method, we add a switch updating model to keep $\hat{\theta}_k$ feasible after performing the SPSA step in each iteration. In other words, when the current point is feasible, it runs one step based on the SPSA algorithm, which will cost two loss function measurements; otherwise, it runs one step based on the gradient of one of the infeasible constraints. We propose the SU algorithm for solving problem (1.1) below. We present the algorithm in two parts. The first, Algorithm 1(a), is the driver that updates the estimate of theta based on new measurements $y(\theta)$. Algorithm 1(a) calls the second part, Algorithm 1(b), as needed, to ensure that the constraints are met. Once constraints are met, the estimate for θ is returned to Algorithm 1(a) for updating with new function measurements.

In the SU algorithm, t is the counter of the total number of iterations, k is the counter of K—the number of iterations where we measure $L(\theta)$, and $\mathcal{M}$ is the operator of Algorithm 1(b) to meet the constraints. We sample Δ_k based on some distribution; a commonly adopted one is the ± 1 Bernoulli distribution for the components of Δ_k in [13].

The switch updating model $\mathcal{M}$ outputs feasible iterates based on $\hat{\theta}_k$. In this model, $\tilde{a}_t$ is a decaying sequence with an adjustable lower bound $a_k/2^{\beta} \to 0$ as $\beta \to \infty$. It is a special case that $\tilde{a}_t = a_k$ when $\beta = 0$, which is a simple and convenient choice to be applied in most situations. This corresponds to a s̀tair-stepd́ecay of a reduced step size at each entry into Algorithm 1(b), but a constant step size in Algorithm 1(b). But in some settings, we need $\beta > 0$ to make sure $\tilde{\theta}_t \in \Theta$ after finite number of iterations.

For each iteration of the SU algorithm, SPSA steps based on $y(\theta)$ contribute to attaining descent, while gradient steps based on $q_i(\theta)$ are used to ensure feasibility. The combination of them shown in the SU algorithm provides convergence results to the exact solution of problem (1.1) with feasibility guaranteed when the number of iterations goes to infinity.

Algorithm 1(a) The SPSA-based Switch Updating Algorithm

```
Input: an initial point $\hat{\theta}_0$, $\tilde{\theta}_0 = \hat{\theta}_0$, $a > 0$, $A \geq 0$, $c > 0$, $\alpha \in (0.5, 1)$, $\gamma \in [\alpha/6, \alpha - 0.5)$, $\beta \geq 0$,
  $t = 0$, $k = 0$, $K > 0$ (if known), the switch updating model $\mathcal{M}$.
  $a_0 = a/(1+A)^\alpha$, $\hat{\theta}_0 = \mathcal{M}(\hat{\theta}_0, a_0, \beta, t)$
  while $k \leq K$ do
    $a_k = a/(k+1+A)^\alpha$, $c_k = c/(k+1)^\gamma$
    $\hat{g}_k(\hat{\theta}_k) = [y(\hat{\theta}_k + c_k\Delta_k) - y(\hat{\theta}_k - c_k\Delta_k)]/2c_k\Delta_k$
    $\hat{\theta}_{k+1} = \hat{\theta}_k - \tilde{a}_t\hat{g}_k(\hat{\theta}_k)$, $\tilde{\theta}_{t+1} = \hat{\theta}_{k+1}$
    $t = t + 1$, $k = k + 1$, $\hat{\theta}_k = \mathcal{M}(\hat{\theta}_k, a_k, \beta, t)$
  end while
Output: $\hat{\theta}_K \in \Theta$.
```

Algorithm 1(b) The Switch Updating Model $\mathcal{M}$

```
Input: $\hat{\theta}_k$, $a_k$, $\beta$ and $t$.
  set $k' = 0$
  while $\hat{\theta}_k \notin \Theta$ do
    for $i = 1, 2, ..., m$ do
      if $q_i(\hat{\theta}_k) > 0$ then
        $\tilde{a}_t = a_k(t+1)^\beta/(t+k'+1)^\beta$
        $\hat{\theta}_k = \hat{\theta}_k - \tilde{a}_t\nabla q_i(\hat{\theta}_k)$, $\tilde{\theta}_{t+1} = \hat{\theta}_k$
        $t = t + 1$, $k' = k' + 1$
        break
      end if
    end for
  end while
Output: $\hat{\theta}_k \in \Theta$.
```

3 Convergence Analysis

In the SU algorithm, since each value of k is related to one or several values of t, we define $k = \kappa(t)$ as a mapping. Then we define function $\mathcal{L}(\theta)$:

$$\mathcal{L}(\theta) = \begin{cases} q_1(\theta) & q_1(\theta) > 0, \\ q_2(\theta) & q_1(\theta) \leq 0, q_2(\theta) > 0, \\ q_3(\theta) & q_1(\theta), q_2(\theta) \leq 0, q_3(\theta) > 0, \\ \cdots\cdots & \\ L(\theta) & q_i(\theta) \leq 0, i = 1, 2, ..., m. \end{cases} \tag{3.1}$$

For the SU algorithm we have:

$$\tilde{\theta}_{t+1} = \tilde{\theta}_t - \tilde{a}_t \hat{h}_t(\tilde{\theta}_t). \tag{3.2}$$

In this updating formula, we have:

$$\hat{h}_t(\tilde{\theta}_t) = \begin{cases} \nabla q_1(\tilde{\theta}_t) & q_1(\tilde{\theta}_t) > 0, \\ \nabla q_2(\tilde{\theta}_t) & q_1(\tilde{\theta}_t) \leq 0, q_2(\tilde{\theta}_t) > 0, \\ \nabla q_3(\tilde{\theta}_t) & q_1(\tilde{\theta}_t), q_2(\tilde{\theta}_t) \leq 0, q_3(\tilde{\theta}_t) > 0, \\ \cdots\cdots & \\ \hat{g}_{\kappa(t)}(\tilde{\theta}_t) & q_i(\tilde{\theta}_t) \leq 0, i = 1, 2, ..., m, \end{cases} \tag{3.3}$$

where

$$\begin{aligned} \hat{g}_{\kappa(t)}(\tilde{\theta}_t) &= \frac{[y(\tilde{\theta}_t^+) - y(\tilde{\theta}_t^-)]}{2c_{\kappa(t)}\Delta_{\kappa(t)}}, \\ \tilde{\theta}_t^+ &= \tilde{\theta}_t + c_{\kappa(t)}\Delta_{\kappa(t)}, \tilde{\theta}_t^- = \tilde{\theta}_t - c_{\kappa(t)}\Delta_{\kappa(t)}. \end{aligned} \tag{3.4}$$

Then we reformulate the updating formula as:

$$\tilde{\theta}_{t+1} = \tilde{\theta}_t - \tilde{a}_t h(\tilde{\theta}_t) - \tilde{a}_t b_t - \tilde{a}_t e_t. \tag{3.5}$$

Here, b_t represents the bias in $\hat{h}_t(\tilde{\theta}_t)$, and e_t denotes the noise term. Let $g(\theta) = \nabla L(\theta)$, $\varepsilon_{\kappa(t)}^+ = \varepsilon(\tilde{\theta}_t^+)$ and $\varepsilon_{\kappa(t)}^- = \varepsilon(\tilde{\theta}_t^-)$. We show each term in (3.5) as:

$$h(\tilde{\theta}_t) = \begin{cases} \nabla q_1(\tilde{\theta}_t) & q_1(\tilde{\theta}_t) > 0, \\ \nabla q_2(\tilde{\theta}_t) & q_1(\tilde{\theta}_t) \leq 0, q_2(\tilde{\theta}_t) > 0, \\ \nabla q_3(\tilde{\theta}_t) & q_1(\tilde{\theta}_t), q_2(\tilde{\theta}_t) \leq 0, q_3(\tilde{\theta}_t) > 0, \\ \cdots\cdots & \\ g(\tilde{\theta}_t) & q_i(\tilde{\theta}_t) \leq 0, i = 1, 2, ..., m, \end{cases} \tag{3.6}$$

$$\begin{aligned} b_t &= \mathbb{E}[\hat{h}_t(\tilde{\theta}_t)|\tilde{\theta}_t] - h(\tilde{\theta}_t) \\ &= \begin{cases} \bar{g}_{\kappa(t)}(\tilde{\theta}_t) - g(\tilde{\theta}_t) & q_i(\tilde{\theta}_t) \leq 0, i = 1, 2, ..., m, \\ 0 & \text{otherwise}, \end{cases} \end{aligned} \tag{3.7}$$

$$
\begin{aligned}
e_t &= \hat{h}_t(\tilde{\theta}_t) - \mathbb{E}[\hat{h}_t(\tilde{\theta}_t)|\tilde{\theta}_t] \\
&= \begin{cases} \hat{g}_{\kappa(t)}(\tilde{\theta}_t) - \bar{g}_{\kappa(t)}(\tilde{\theta}_t) & q_i(\tilde{\theta}_t) \leq 0, i = 1, 2, ..., m, \\ 0 & \text{otherwise,} \end{cases}
\end{aligned} \tag{3.8}
$$

where

$$
\bar{g}_{\kappa(t)}(\tilde{\theta}_t) = \mathbb{E}[\hat{g}_{\kappa(t)}(\tilde{\theta}_t)|\tilde{\theta}_t] = \frac{[L(\tilde{\theta}_t^+) - L(\tilde{\theta}_t^-)]}{2c_{\kappa(t)}\Delta_{\kappa(t)}}. \tag{3.9}
$$

And the noise term could be simplified as:

$$
e_t = \begin{cases} \frac{\varepsilon_{\kappa(t)}}{2c_{\kappa(t)}\Delta_{\kappa(t)}} & q_i(\tilde{\theta}_t) \leq 0, i = 1, 2, ..., m, \\ 0 & \text{otherwise,} \end{cases} \tag{3.10}
$$

where

$$
\varepsilon_{\kappa(t)} = \varepsilon^+_{\kappa(t)} - \varepsilon^-_{\kappa(t)}. \tag{3.11}
$$

Based on the convergence result for the SPSA algorithm analyzed in [1], and according to the updating formula (3.5), we make the following assumptions to ensure almost sure convergence for the SU algorithm:

Assumption A. $a_k > 0$, $c_k > 0$, $a_k \to 0$, $c_k \to 0$, $\sum_{k=0}^{\infty} a_k = \infty$, $\sum_{k=0}^{\infty} a_k^2/c_k^2 < \infty$. $\forall t$, $\exists t' < \infty$ s.t. $\kappa(t + t') > \kappa(t)$.

Assumption B. *$\forall k, t, j$, Δ_{kj} are i.i.d. and symmetrically distributed about 0 and $\exists \delta, \alpha_0, \alpha_1, \alpha_2 > 0$ s.t.$\mathbb{E}[|\varepsilon_k^{\pm}|^{2+\delta}] \leq \alpha_0$, $\mathbb{E}[|L(\tilde{\theta}_t^{\pm})|^{2+\delta}|\tilde{\theta}_t \in \Theta] \leq \alpha_1$, $|\Delta_{kj}| \leq \alpha_2$, and $\mathbb{E}[|\Delta_{kj}|^{-2-\delta}] \leq \alpha_3$.*

Assumption C. *$\forall t, \|\tilde{\theta}_t\| < \infty$ almost surely.*

Assumption D. *θ^* is an asymptotically stable solution of the differential equation $dx(s)/ds = -h(x)$:*

$$
\frac{dx(s)}{ds} = \begin{cases} -\nabla q_1(x) & q_1(x) > 0, \\ -\nabla q_2(x) & q_1(x) \leq 0, q_2(x) > 0, \\ -\nabla q_3(x) & q_1(x), q_2(x) \leq 0, q_3(x) > 0, \\ \cdots\cdots & \\ -g(x) & q_i(x) \leq 0, i = 1, 2, ..., m. \end{cases} \tag{3.12}
$$

Assumption E. *Let $D(\theta^*) = \{x_0 | \lim_{s\to\infty} x(s|x_0) = \theta^*\}$ where $x(s|x_0)$ denotes the solution to the differential equation (3.12) based on initial conditions x_0. Then there exists a compact set $S \subseteq D(\theta^*)$ such that $\tilde{\theta}_t \in S$ infinitely often for almost all sample points.*

For Assumption A, since the gain sequence related to the gradients of the constraint functions does not decay to 0 before reaching the next feasible point, it is required that their values should be small enough to ensure attaining feasibility in finite number of iterations. This is equivalent that a_0 is not extremely large and β is sufficiently large to avoid diverging, which is not difficult to be satisfied. Assumption D constructs the ODE equation based on not only the gradient of the objective loss function but also the gradients of all the constraint functions. Although $h(\theta)$ defined for the SU algorithm is not a continuous function on $\mathbb{R}^p$, it would not be too difficult to let θ^* be an asymptotically stable solution for the ODE shown in Assumption D, and $\nabla L(\theta^*)$, $\nabla q_i(\theta^*)$ $(i = 1, ..., m)$ being nonzero can make it even more stable than under condition $\nabla L(\theta^*) = 0$ in unconstrained problem cases. Assumption D is crucial to Proposition 3.1, and as a special case to satisfy it, it is similar to requiring the objective loss function and all the constraint functions to be partly strictly convex.

Proposition 3.1 given below shows almost sure convergence for the SU algorithm.

Proposition 3.1. *Suppose that Assumptions A–E hold. Then when $t \to \infty$, $\hat{\theta}_k \to \theta^*$ almost surely in the SU algorithm.*

Proof. According to Assumption A, whenever $\tilde{\theta}_t$ is infeasible in the SU algorithm, we will attain a feasible point for problem (1.1) in finite numbers of iterations. This indicates that in the SU algorithm, $k = \kappa(t) \to \infty$ as $t \to \infty$.

According to Assumption A, B and Lemma 1 in [1], we can directly know:

$$\|b_t\| < \infty \ \forall t \text{ and } b_t \to 0 \text{ almost surely.} \tag{3.13}$$

In as much as $\{\sum_{i=l}^{n} \tilde{a}_i e_i\}_{n \geq l}$ is a martingale sequence, and $\mathbb{E}[e_i^T e_j] = \mathbb{E}[e_i^T \mathbb{E}[e_j|\tilde{\theta}_j]]$ $\forall i < j$, we have:

$$\begin{aligned} P(\sup_{n \geq l} \| \sum_{i=l}^{n} \tilde{a}_i e_i \| \geq \eta) &\leq \eta^{-2} \mathbb{E}[\| \sum_{i=l}^{n} \tilde{a}_i e_i \|^2] \\ &= \eta^{-2} \sum_{i=l}^{n} \tilde{a}_i^2 \mathbb{E}[\|e_i^2\|]. \end{aligned} \tag{3.14}$$

To establish an upper bound for $\mathbb{E}\|e_t^2\|$, as $e_t = 0$ when $\tilde{\theta}_t$ is infeasible, we only need to consider the case that $\tilde{\theta}_t$ is feasible. According to Hölder's theorem and Assumption B, we can obtain:

$$\begin{aligned} \mathbb{E}[|\varepsilon_{\kappa(t)}^{\pm}|^2] &\leq \mathbb{E}[1^{\frac{2+\delta}{1+\delta}}]^{\frac{1+\delta}{2+\delta}} \mathbb{E}[|\varepsilon_{\kappa(t)}^{\pm}|^{2+\delta}]^{\frac{1}{2+\delta}} \\ &= \mathbb{E}[|\varepsilon_{\kappa(t)}^{\pm}|^{2+\delta}]^{\frac{1}{2+\delta}} \\ &\leq \alpha_0^{\frac{1}{2+\delta}} = \alpha_0'. \end{aligned} \tag{3.15}$$

By the similar process, we can find that $\mathbb{E}[L(\tilde{\theta}_t^{\pm})^2|\tilde{\theta}_t \in \Theta]$ and $\mathbb{E}[\Delta_{\kappa(t)j}^{-2}]$ are also both bounded. Let the upper bound for them be α_1' and α_3', then we have:

$$\begin{aligned} \mathbb{E}[\hat{g}_{\kappa(t)j}(\tilde{\theta}_t)^2] \leq & \frac{1}{4} \mathbb{E}[(L(\tilde{\theta}_t^+) - L(\tilde{\theta}_t^-) + \varepsilon_{\kappa(t)}^+ - \varepsilon_{\kappa(t)}^-)^2] \\ & \mathbb{E}[(c_{\kappa(t)} \Delta_{\kappa(t)j})^{-2}] \\ \leq & 2(\alpha_1' + \alpha_0')\alpha_3' c_{\kappa(t)}^{-2}. \end{aligned} \tag{3.16}$$

Thus, when $\tilde{\theta}_t$ is feasible, for $\mathbb{E}[\|e_t\|^2]$ we have:

$$\begin{aligned} \mathbb{E}[\|e_t\|^2] &\leq p \max_{1 \leq j \leq p} \mathbb{E}[e_{tj}^2] \\ &= p \max_{1 \leq j \leq p} \mathbb{E}[(\hat{g}_{\kappa(t)j}(\tilde{\theta}_t) - \mathbb{E}[\hat{g}_{\kappa(t)j}(\tilde{\theta}_t)])^2] \\ &= p \max_{1 \leq j \leq p} [\mathbb{E}[\hat{g}_{\kappa(t)j}(\tilde{\theta}_t)^2] - (\mathbb{E}[\hat{g}_{\kappa(t)j}(\tilde{\theta}_t)])^2] \\ &\leq p \max_{1 \leq j \leq p} \mathbb{E}[\hat{g}_{\kappa(t)j}(\tilde{\theta}_t)^2] \\ &\leq 2p(\alpha_1' + \alpha_0')\alpha_3' c_{\kappa(t)}^{-2}. \end{aligned} \tag{3.17}$$

By (3.14), (3.17) and Assumption A, we have:

$$\begin{aligned}&\lim_{l\to\infty} P(\sup_{n\geq l} \| \sum_{i=l}^{n} \tilde{a}_i e_i \| \geq \eta)\\ \leq& 2p\eta^{-2}(\alpha_1' + \alpha_0')\alpha_3' \lim_{l\to\infty} \sum_{r=\kappa(l)}^{\kappa(n)} a_r^2 c_r^{-2}\\ =&0. \end{aligned} \tag{3.18}$$

Therefore, according to Assumptions A, C, D, E, (3.13) and (3.18), and based on Theorem 2.3.1 in [9], we can finally reach the result that when $t \to \infty$, $\tilde{\theta}_t \to \theta^*$ and $\hat{\theta}_k \to \theta^*$ almost surely in the SU algorithm.

□

4 Convergence Rate Analysis

This part shows the convergence rate of the SU algorithm. We will give an upper bound for the asymptotic convergence rate performance of our algorithm in the big-O sense.

We use a mean gradient-like expression $\bar{h}(\tilde{\theta}_t) = \mathbb{E}[\hat{h}_t(\tilde{\theta}_t)|\tilde{\theta}_t]$ for our analysis in this part, which is:

$$\bar{h}(\tilde{\theta}_t) = \begin{cases} \nabla q_1(\tilde{\theta}_t) & q_1(\tilde{\theta}_t) > 0, \\ \nabla q_2(\tilde{\theta}_t) & q_1(\tilde{\theta}_t) \leq 0, q_2(\tilde{\theta}_t) > 0, \\ \nabla q_3(\tilde{\theta}_t) & q_1(\tilde{\theta}_t), q_2(\tilde{\theta}_t) \leq 0, q_3(\tilde{\theta}_t) > 0, \\ \cdots\cdots & \\ \bar{g}_{\kappa(t)}(\tilde{\theta}_t) & q_i(\tilde{\theta}_t) \leq 0, i = 1, 2, ..., m, \end{cases} \tag{4.1}$$

As an example, we assume that each element of $\Delta_{\kappa(t)} \in \mathbb{R}^p$ here follows ±1 Bernoulli distribution. Then let $\sum_{\Delta_{\kappa(t)}}$ be the summation of all the possible $\Delta_{\kappa(t)}$, then we have:

$$\bar{g}_{\kappa(t)}(\tilde{\theta}_t) = \frac{1}{2^p} \sum\nolimits_{\Delta_{\kappa(t)}} \frac{L(\tilde{\theta}_t^+) - L(\tilde{\theta}_t^-)}{2c_{\kappa(t)}} \Delta_{\kappa(t)}^{-1}. \tag{4.2}$$

Then we make the following assumptions to ensure the upper bound of the convergence rate for the SU algorithm:

Assumption F. *The components of $\Delta_{\kappa(t)}$ are i.i.d. random variables and $0 < \Delta_{\kappa(t)}^{-T}\Delta_{\kappa(t)}^{-1} \leq d < \infty$.*

Assumption G. *For all t, $\mathbb{E}[\varepsilon_{\kappa(t)}^{+} - \varepsilon_{\kappa(t)}^{-}|\tilde{\theta}_{\kappa(t)}, \Delta_{\kappa(t)}] = 0$ almost surely, and $\mathrm{var}(\varepsilon_{\kappa(t)}^{\pm})$ is uniformly bounded in $\kappa(t)$.*

Assumption H. *$\mathbb{E}[(L(\tilde{\theta}_t^+) - L(\tilde{\theta}_t^-))^2|\tilde{\theta}_t]$ and $\|\nabla q_i(\tilde{\theta}_t)^2\|$ are uniformly bounded for all t.*

Assumption I. *When $\tilde{\theta}_t$ is infeasible, it will need not more than M iterations to get into the feasible set.*

Assumption J. *There exists $\mu > 0$ such that $(\tilde{\theta}_t - \theta^*)^T\bar{h}(\tilde{\theta}_t) - \mu(\tilde{\theta}_t - \theta^*)^T(\tilde{\theta}_t - \theta^*) \geq 0$ for all t.*

For Assumption H, it means that the sequence of $\{\tilde{\theta}_t\}$ should not jump to the "far away" area so often, which is a natural consequence if $\|\tilde{\theta}_t\|$ is almost surely bounded. For Assumption I, it is easy to satisfy if, to avoid diverging, we select a value of a_0 that is not too large. For Assumption J, it is similar to state that $\mathcal{L}(\theta)$ is quasi-strongly convex if $\bar{g}_{\kappa(t)}(\tilde{\theta}_t) = g(\tilde{\theta}_t)\ \forall t$. We say $f(\theta)$ is a μ'-quasi-strongly convex function if with θ^* as its unique optimal solution, there exists $\mu' > 0$ such that for any θ:

$$f(\theta^*) \geq f(\theta) + \langle \nabla f(\theta), \theta^* - \theta \rangle + \frac{\mu'}{2}\|\theta - \theta^*\|^2. \tag{4.3}$$

Specifically, since $\bar{h}(\tilde{\theta}_t)$ is composed of several parts as shown in (4.1): $\nabla q_i(\tilde{\theta}_t)$ $(i = 1, ..., m)$ and $\bar{g}_{\kappa(t)}(\tilde{\theta}_t)$, Assumption J actually requires all the constraint functions and the loss function to be partly quasi-strongly convex, with θ^* being the same optimal solution for them.

Proposition 4.1 given below shows the upper bound of the convergence rate for the SU algorithm.

Proposition 4.1. *Suppose that Assumptions F–J hold. Then the upper bound of the convergence rate for the SU algorithm is $O(1/t^{\alpha})$.*

Proof. First we consider the situation that $\tilde{\theta}_t$ is infeasible. In this case $\tilde{a}_t = \frac{a_k(t+1)^{\beta}}{(t+k'+1)^{\beta}}$ and $\hat{h}_t(\tilde{\theta}_t) = \nabla q_i(\tilde{\theta}_t)$ for some $i = 1, ..., m$, then we have:

$$\tilde{\theta}_{t+1} = \tilde{\theta}_t - \frac{a_k(t+1)^{\beta}}{(t+k'+1)^{\beta}}\nabla q_i(\tilde{\theta}_t). \tag{4.4}$$

Subtracting θ^* from both sides and calculating the norm squared, we get:

$$\begin{aligned}\|\tilde{\theta}_{t+1} - \theta^*\|^2 =& \|\tilde{\theta}_t - \theta^*\|^2 - 2\tilde{a}_t(\tilde{\theta}_t - \theta^*)^T \nabla q_i(\tilde{\theta}_t)\\ &+ \tilde{a}_t^2\|\nabla q_i(\tilde{\theta}_t)\|^2. \end{aligned}\tag{4.5}$$

According to Assumption H, since $\|\nabla q_i(\tilde{\theta}_t)\|^2$ is uniformly bounded, there exists $B > 0$ such that $\|\nabla q_i(\tilde{\theta}_t)\|^2 \leq B$. Then we have:

$$\begin{aligned}\|\tilde{\theta}_{t+1} - \theta^*\|^2 =& \|\tilde{\theta}_t - \theta^*\|^2 - 2\tilde{a}_t(\tilde{\theta}_t - \theta^*)^T \nabla q_i(\tilde{\theta}_t)\\ &+ \tilde{a}_t^2\|\nabla q_i(\tilde{\theta}_t)\|^2\\ \leq& (1 - 2\mu\tilde{a}_t)\|\tilde{\theta}_t - \theta^*\|^2 + \tilde{a}_t^2 B\\ =& \left[1 - \frac{2\mu a_k(t+1)^{\beta}}{(t+k'+1)^{\beta}}\right]\|\tilde{\theta}_t - \theta^*\|^2\\ &+ \frac{a_k^2(t+1)^{2\beta}}{(t+k'+1)^{2\beta}}B\\ \leq& \left(1 - \frac{2\mu a_k}{2^{\beta}}\right)\|\tilde{\theta}_t - \theta^*\|^2 + a_k^2 B. \end{aligned}\tag{4.6}$$

Similarly, when $\tilde{\theta}_t$ is feasible with $k = \kappa(t)$, now $\tilde{a}_t = a_{\kappa(t)} = a_k$, and we have:

$$\tilde{\theta}_{t+1} = \tilde{\theta}_t - \tilde{a}_t\hat{g}_{\kappa(t)}(\tilde{\theta}_t) = \tilde{\theta}_t - a_k\hat{g}_{\kappa(t)}(\tilde{\theta}_t). \tag{4.7}$$

Since $\bar{h}(\tilde{\theta}_t) = \mathbb{E}[\hat{h}_t(\tilde{\theta}_t)]$, we still subtract θ^* and calculate the norm squared, and take the expectation condition on $\tilde{\theta}_t$. Then we get the inequality similar as above:

$$\begin{aligned}\mathbb{E}[\|\hat{\theta}_{t+1} - \theta^*\|^2|\tilde{\theta}_t] =& \|\tilde{\theta}_t - \theta^*\|^2 + a_k^2\mathbb{E}[\|\hat{g}_{\kappa(t)}(\tilde{\theta}_t)|\tilde{\theta}_t\|^2] \\ &- 2a_k(\tilde{\theta}_t - \theta^*)^T\mathbb{E}[\hat{g}_{\kappa(t)}(\tilde{\theta}_t)|\tilde{\theta}_t] \\ \le& (1 - 2\mu a_k)\|\tilde{\theta}_t - \theta^*\|^2 \\ &+ a_k^2\mathbb{E}[\|\hat{g}_{\kappa(t)}(\tilde{\theta}_t)|\tilde{\theta}_t\|^2]. \end{aligned} \tag{4.8}$$

According to Assumption G and H, $\mathbb{E}[(L(\tilde{\theta}_t^+) - L(\tilde{\theta}_t^-))^2|\tilde{\theta}_t] + \mathbb{E}[(\varepsilon_{\kappa(t)}^+ - \varepsilon_{\kappa(t)}^-)^2|\tilde{\theta}_t]$ is upper bounded. Let us assume that its upper bound is b, therefore we have:

$$\begin{aligned}\mathbb{E}[\hat{g}_{\kappa(t)}(\tilde{\theta}_t)|\tilde{\theta}_t\|^2] =& \mathbb{E}[(L(\tilde{\theta}_t^+) - L(\tilde{\theta}_t^-))^2\Delta_{\kappa(t)}^{-T}\Delta_{\kappa(t)}^{-1}|\tilde{\theta}_t] \\ &+ \mathbb{E}[(\varepsilon_{\kappa(t)}^+ - \varepsilon_{\kappa(t)}^-)^2\Delta_{\kappa(t)}^{-T}\Delta_{\kappa(t)}^{-1}|\tilde{\theta}_t] \\ \le& pd(\mathbb{E}[(L(\tilde{\theta}_t^+) - L(\tilde{\theta}_t^-))^2|\tilde{\theta}_t] \\ &+ \mathbb{E}[(\varepsilon_{\kappa(t)}^+ - \varepsilon_{\kappa(t)}^-)^2|\tilde{\theta}_t]) \\ \le& pdb. \end{aligned} \tag{4.9}$$

Take (4.9) into (4.8) and take the expectation, we have:

$$\mathbb{E}[\|\tilde{\theta}_{t+1} - \theta^*\|^2] \le (1 - 2\mu a_k)\mathbb{E}[\|\tilde{\theta}_t - \theta^*\|^2] + a_k^2 pdb. \tag{4.10}$$

According to Assumption I, we have $k \le t \le Mk$. Considering the inequalities (4.6) and (4.10), let $\mu^* = \min\{\mu/2^\beta, 1/(4a_0)\}$ and $B^* = \max\{B, pdb\}$, then for any iteration t in the whole process, we have $1 - 2\mu^* a_t > 0$ and:

$$\begin{aligned}\mathbb{E}[\|\tilde{\theta}_{t+1} - \theta^*\|^2] &\le (1 - 2\mu^* a_k)\mathbb{E}[\|\tilde{\theta}_t - \theta^*\|^2] + a_k^2 B^* \\ &\le (1 - 2\mu^* a_t)\mathbb{E}[\|\tilde{\theta}_t - \theta^*\|^2] + a_t^2 M^\alpha B^*. \end{aligned} \tag{4.11}$$

Using the proof in [14], we get the following inequality:

$$\begin{aligned}&\mathbb{E}[\|\tilde{\theta}_t - \theta^*\|^2] \\ \le& \exp\left\{\frac{2\mu^* a[(1+A)^{1-\alpha} - (1+A+t)^{1-\alpha}]}{1-\alpha}\right\} \\ &\left[\|\tilde{\theta}_0 - \theta^*\|^2 - \frac{T(t,\alpha)}{(1+A)^\alpha}\right] + \frac{T(t,\alpha)}{(1+A+t)^\alpha}, \end{aligned} \tag{4.12}$$

where

$$\begin{aligned} T(t,\alpha) &= \frac{M^\alpha B^* a^2 C(\alpha)}{2\mu^* a - \alpha/[1+A+f(t)]^{1-\alpha}}, \\ C(\alpha) &= \exp\left\{\frac{2\mu^* a}{1-\alpha}\left[(2+A)^{1-\alpha} - (1+A)^{1-\alpha}\right]\right\}\left(\frac{2+A}{1+A}\right)^{2\alpha}, \\ f(t) &= \left\{\frac{\int_0^t (1+A+x)^{-2\alpha}\exp\{\frac{2\mu^* a(1+A+x)^{1-\alpha}}{1-\alpha}\}dx}{\int_0^t (1+A+x)^{-1-\alpha}\exp\{\frac{2\mu^* a(1+A+x)^{1-\alpha}}{1-\alpha}\}dx}\right\}^{\frac{1}{1-\alpha}} - 1 - A. \end{aligned} \tag{4.13}$$

Then we can rewrite the equation of $f(t)$ as:

$$\frac{1}{[1+A+f(t)]^{1-\alpha}} = \frac{\int_0^t (1+A+x)^{-1-\alpha} \exp\{\frac{2\mu^* a(1+A+x)^{1-\alpha}}{1-\alpha}\}dx}{\int_0^t (1+A+x)^{-2\alpha} \exp\{\frac{2\mu^* a(1+A+x)^{1-\alpha}}{1-\alpha}\}dx}. \tag{4.14}$$

By L'Hôpital's rule, when $t \to \infty$, we have:

$$\begin{aligned} &\lim_{t\to\infty} \frac{1}{[1+A+f(t)]^{1-\alpha}} \\ =& \lim_{t\to\infty} \frac{\int_0^t (1+A+x)^{-1-\alpha} \exp\{\frac{2\mu^* a(1+A+x)^{1-\alpha}}{1-\alpha}\}dx}{\int_0^t (1+A+x)^{-2\alpha} \exp\{\frac{2\mu^* a(1+A+x)^{1-\alpha}}{1-\alpha}\}dx} \\ =& \lim_{t\to\infty} (1+A+t)^{\alpha-1} \\ =&0. \end{aligned} \tag{4.15}$$

From (4.15), we know that as $t \to \infty$, $f(t) \to \infty$, then we can get $T(t,\alpha) \to \frac{M^\alpha B^* aC(\alpha)}{2\mu^*}$, which is a constant. When t is large enough, the effect of A can be ignored, thus the convergence rate of the second term in RHS of (4.12) is $O(1/t^\alpha)$. For the first term in its RHS, we only need to consider the exponential part, and it is obvious that this part goes to 0 faster than the second term. Therefore, we can finally get the upper bound of $\mathbb{E}[\|\hat{\theta}_t - \theta^*\|^2]$ as $O(1/t^\alpha)$.

□

5 Asymptotic Result for Number of Loss Function Measurements

In this section, we are going to gain some insight into the perspective of the number of loss function measurements. It is known that one of the advantages for the SPSA algorithm over the finite difference stochastic approximation (FDSA) algorithm is that the SPSA algorithm only requires 2 loss function measurements in each iteration, while the FDSA algorithm needs $2p$ loss function measurements. Thus, compared to the FDSA algorithm, the SPSA algorithm increases the efficiency of loss function measurements by p times. For our SU algorithm, when $\tilde{\theta}_t$ is infeasible under current iteration, we use $\nabla q_i(\tilde{\theta}_t)$ in our updating formula instead of conducting an SPSA step based on two measurements of $L(\theta)$. In as much as we do not expect to measure loss function in every iteration, then compared to the SPSA algorithm using penalty ideas, fewer loss function measurements will be made under the same number of total iterations. Here, we give an asymptotic result for the ratio of iterations that necessitates loss function measurements in the SU algorithm.

According to Karush-Kuhn-Tucker (KKT) conditions, we define the set of KKT points as the asymptotically stable optimal solutions for problem (1.1).

Definition 5.1. *The set of KKT points are defined as:* $\{\theta \mid \forall i \ q_i(\theta) \le 0, \exists \lambda_i \ge 0 \text{ s.t.} \lambda_i q_i(\theta) = 0, \frac{dL(\theta)}{d\theta} + \sum_{i=1}^m \lambda_i \frac{dq_i(\theta)}{d\theta} = 0\}$.

To further let KKT points necessarily be the optimal solutions for problem (1.1), some type of constraint qualification should be satisfied. We then define a common type of constraint qualification called linear independence constraint qualification (LICQ) condition.

Definition 5.2. *Let set* $A(\theta) = \{i | q_i(\theta) = 0, i = 1, ..., m\}$. *When* $\nabla q_i(\theta)$ $(i \in A(\theta))$ *are linearly independent at* θ*, the LICQ condition is satisfied at* θ *for problem* (1.1).

Moreover, if $A(\theta) = \phi$, then the LICQ condition is automatically satisfied at θ.

Proposition 5.3. *Suppose that Assumptions A–E hold. Let θ^* be a KKT point and the optimal solution for problem* (1.1)*, and λ_i $(i = 1, ..., m)$ be the Lagrange multipliers corresponding to θ^* in Definition 5.1. Assume the LICQ condition is satisfied at θ^* for problem* (1.1)*. Then when $t \to \infty$, the proportion of the iterations that make loss function measurements in the SU algorithm is $\frac{1}{1+\sum_{i=1}^{m} \lambda_i}$.*

Proof. We know that the optimal solution θ^* for our problem (1.1) is a KKT point for it. Due to Assumption A, $k = \kappa(t) \to \infty$ as $t \to \infty$. Then according to Proposition 3.1, when $t \to \infty$, $\hat{\theta}_k \to \theta^*$ almost surely in the SU algorithm. Considering $l = k^{\frac{1+\alpha}{2}} \to \infty$ iterations after the t-th iteration with $k = \kappa(t)$, we have:

$$\lim_{t\to\infty} \tilde{\theta}_t = \lim_{t\to\infty} [\tilde{\theta}_t - \sum_{i=0}^{l-1} \tilde{a}_{t+i} \hat{h}_{t+i}(\tilde{\theta}_{t+i})]. \tag{5.1}$$

Since $b_t = \mathbb{E}[\hat{h}_t(\tilde{\theta}_t)|\tilde{\theta}_t] - h(\tilde{\theta}_t) \to 0$ and $e_t = \hat{h}_t(\tilde{\theta}_t) - \mathbb{E}[\hat{h}_t(\tilde{\theta}_t)|\tilde{\theta}_t] \to 0$ as $t \to \infty$, we have $\hat{h}_{t+i}(\tilde{\theta}_{t+i}) \to h(\tilde{\theta}_{t+i})$. Note that since $h(\theta)$ is not continuous at θ^*, we cannot get $h(\tilde{\theta}_{t+i}) \to h(\theta^*)$ as $\tilde{\theta}_{t+i} \to \theta^*$. Then (5.1) becomes:

$$\theta^* = \theta^* - \lim_{t\to\infty} \sum_{i=0}^{l-1} \tilde{a}_{t+i} h(\tilde{\theta}_{t+i}). \tag{5.2}$$

Since $\lim_{t\to\infty} \frac{l}{t} \leq \lim_{t\to\infty} \frac{l}{\kappa(t)} = 0$, we have $\lim_{t\to\infty} \tilde{a}_t = \lim_{t\to\infty} \tilde{a}_{t+i}$ when $0 \leq i < l$. Then we get:

$$\begin{aligned} \lim_{t\to\infty} \sum_{i=0}^{l-1} \tilde{a}_{t+i} h(\tilde{\theta}_{t+i}) &= 0, \\ \lim_{t\to\infty} \tilde{a}_t l \frac{\sum_{i=0}^{l-1} h(\tilde{\theta}_{t+i})}{l} &= 0. \end{aligned} \tag{5.3}$$

Since $\lim_{t\to\infty} \tilde{a}_t l > \lim_{t\to\infty} \frac{1}{2^\beta} a_{\kappa(t)} l > 0$, we have:

$$\lim_{t\to\infty} \frac{\sum_{i=0}^{l-1} h(\tilde{\theta}_{t+i})}{l} = 0. \tag{5.4}$$

Assume that in l iterations, we update l^* times based on functional measurements of $L(\theta)$, and l_i $(i = 1, 2, ..., m)$ times based on $\nabla q_i(\theta)$ respectively, then $l = l^* + \sum_{i=1}^{m} l_i$. Denote $g(\theta) = \nabla L(\theta)$, when $t \to \infty$ we have:

$$\lim_{t\to\infty} \frac{l^* g(\theta^*) + \sum_{i=1}^{m} l_i \nabla q_i(\theta^*)}{l} = 0. \tag{5.5}$$

Since the LICQ condition is satisfied at KKT point θ^*, then we define a set $A = \{i | q_i(\theta^*) = 0\}$. For all $i \in A$, $\nabla q_i(\theta^*)$ are linearly independent. Therefore, according to Definition 5.1, there exists a unique set of λ_i $(i = 1, 2, ..., m)$ that $\lambda_i = 0$ if $i \notin A$, and they satisfy:

$$g(\theta^*) + \sum_{i=1}^{m} \lambda_i \nabla q_i(\theta^*) = 0. \tag{5.6}$$

Therefore, we finally have:

$$\lim_{t\to\infty} \frac{l^*}{l} = \lim_{t\to\infty} \frac{l^*}{l^* + \sum_{i=1}^{m} l_i} = \frac{1}{1 + \sum_{i=1}^{m} \lambda_i}. \tag{5.7}$$

This result is satisfied when $t \to \infty$, thus it shows the asymptotic proportion of the iterations that make loss function measurements in the SU algorithm.

□

Proposition 5.3 given above shows that as $t \to \infty$ in the SU algorithm, k/t is a constant value. This also allows the result of Proposition 4.1 to be equivalently based on the number of loss function measurements.

6 Numerical Analysis

6.1 Overview

In this section, we implement the SU algorithm solving two different constrained stochastic optimization problems, with quadratic and quartic loss functions respectively. A main difference is that for quadratic loss functions, the bias $b_t = 0$ for any t, while there usually exists nonzero bias for quartic loss functions during iterations.

We compare our algorithm with the SPSA algorithms based on penalty ideas given in [10]. Basically, by applying penalty ideas, problem (1.1) is constructed as an unconstrained problem in each iteration k as:

$$\min \ L_k(\theta) = L(\theta) + r_k P(\theta), \tag{6.1}$$

where $P : \mathbb{R}^p \to \mathbb{R}^+$ is the penalty function, and the penalty gain sequence $\{r_k\}$ consists of positive values. We consider comparing the SU algorithm with the SPSA algorithms using three different kinds of penalty functions $P(\theta)$: The absolute value penalty (AVP) function, the quadratic penalty (QP) function, and the augmented Lagrange (AL) function.

To initialize the hyper-parameters of these constrained SPSA algorithms, we use the same following gain sequences: $a_k = a/(k+1+A)^\alpha$ and $c_k = c/(k+1)^\gamma$. Standard criteria for the selection of a, A, c, α, γ could be seen from [12]. Furthermore, for the penalty functions, we choose $r_k = r(k+1)^\rho$ as the penalty gain sequence, where ρ represents the growth rate, and it should satisfy $\alpha - \gamma - 2\rho > 0$ and $3\gamma - \alpha/2 + 3\rho/2 > 0$ from [10]. For the AVP function, $r_n = r$ is chosen since the minimum of $L + rP$ is identical to the original problem (6.1) for all $r > \bar{r}$, where $\bar{r}$ is sure to exist according to Proposition 4.3.1 in [15].

For all the algorithms in our numerical experiments, we use the same values of the random generated noises ε_k and the sample Δ_k in the same iteration of each replicate to achieve a fair comparison.

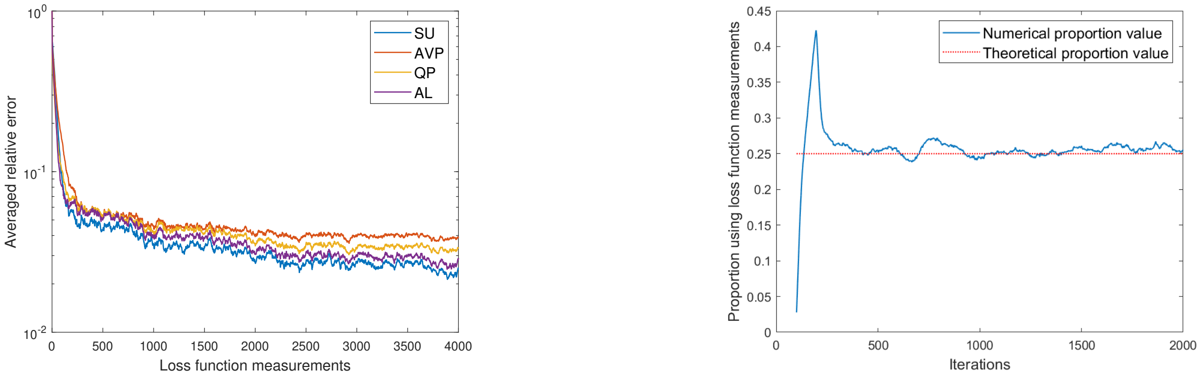
6.2 A Quadratic Example

The first test case we use is a quadratic example, which comes from [10]:

$$\begin{aligned}
\min_{\theta} \ & L(\theta) = t_1^2 + t_2^2 + 2t_3^2 + t_4^2 - 5t_1 - 5t_2 - 21t_3 + 7t_4, \\
\text{s.t.} \ & q_1(\theta) = 2t_1^2 + t_2^2 + t_3^2 + 2t_1 - t_2 - t_4 - 5 \le 0, \\
& q_2(\theta) = t_1^2 + t_2^2 + t_3^2 + t_4^2 + t_1 - t_2 + t_3 - t_4 - 8 \le 0, \\
& q_3(\theta) = t_1^2 + 2t_2^2 + t_3^2 + 2t_4^2 - t_1 - t_4 - 10 \le 0.
\end{aligned} \tag{6.2}$$

In this test example, $\theta = [t_1, t_2, t_3, t_4]^T$. The optimal point $\theta^* = [0, 1, 2, -1]^T$ and the constraints $q_1(\theta)$ and $q_2(\theta)$ are active at θ^*. The Lagrange multiplier at θ^* is $\lambda^* = [2, 1, 0]$. We add i.i.d. noise following normal distribution $N(0,4)$ to the objective function and choose the initial point $\hat{\theta}_0$ as $[-2,-2,-2,-2]^T$, which is outside the feasible set. We set $\alpha = 0.602$, $\gamma = 0.101$, $\beta = 1$, $a = 0.1$, $A = 100$, and $c = 1$. For the QP algorithm, we set $r_k = 2(k+1)^{0.1}$. For the AL algorithm, we set $r_k = (k+1)^{0.1}$ and the initial value of λ_k as $[0,0,0]^T$. For the AVP algorithm, we set $r_k = r = 3.5$.

We generate the results of the averaged relative error $\|\hat{\theta}_K - \theta^*\|/\|\hat{\theta}_0 - \theta^*\|$ based on 50 independent replicates with 4000 loss function measurements ($K = 2000$) in each replicate, and define a non-negative value $Q(\theta) = \sum_{i=1}^{m} \max\{0, q_i(\theta)\}/m$, using the averaged $Q(\hat{\theta}_K)$ to measure the amount of constraint violation of the final solutions. Smaller values of the averaged $Q(\hat{\theta}_K)$ means less infeasibility, while the value of zero indicates all feasible outputs. Besides, we also conduct three pairs of two-sample tests between the SU algorithm and each of the AVP, QP and AL algorithm, and each with the null hypothesis H_0: The averaged relative error of the SU algorithm is larger or equal than the compared algorithm. A smaller p-value means that we are more confident to reject H_0. Additionally, we compute the proportion of the iterations using loss function measurements in the last 100 iterations. Below are the results.



(a) Averaged Relative Error

(b) Asymptotic Proportion

Figure 1: The Quadratic Example

	$\|\hat{\theta}_K - \theta^*\|/\|\hat{\theta}_0 - \theta^*\|$	p-value versus SU	$Q(\hat{\theta}_K)$
SU	0.1374	NA	0
AVP	0.2149	1.1083×10^{-08}	0.4988
QP	0.1847	5.0040×10^{-04}	0.2177
AL	0.1587	0.0638	0.0498

Table 1: Averaged Relative Error, Feasibility
and Two-sample Test Result

6.3 A Quartic Example

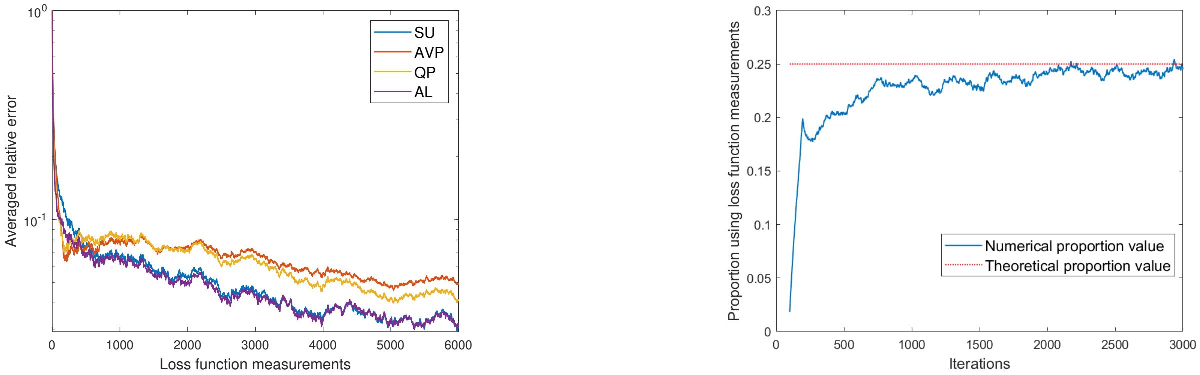
The second test example we use is a quartic case revised from our previous test example (6.2), which is also an example of Exercise 5.5 shown in [12]:

$$\begin{aligned} \min_{\theta}\ L(\theta) =& \sum_{i=1}^{2} t_i^4 + \theta^T B\theta + \theta^T V, \\ \text{s.t. } q_1(\theta) =& 2t_1^2 + t_2^2 + t_3^2 + 2t_1 - t_2 - t_4 - 5 \leq 0, \\ q_2(\theta) =& t_1^2 + t_2^2 + t_3^2 + t_4^2 + t_1 - t_2 + t_3 - t_4 - 8 \leq 0, \\ q_3(\theta) =& t_1^2 + 2t_2^2 + t_3^2 + 2t_4^2 - t_1 - t_4 - 10 \leq 0, \end{aligned} \tag{6.3}$$

where the values of the parameters are:

$$B = \begin{bmatrix} 0 & 0 & 3.5 & 0 \\ 0 & 1 & 0 & -8 \\ 3.5 & 0 & 8 & 0 \\ 0 & -8 & 0 & 5 \end{bmatrix}, V = \begin{bmatrix} -19 \\ -25 \\ -45 \\ 31 \end{bmatrix} + \varepsilon. \tag{6.4}$$

In this test example, $\varepsilon \in \mathbb{R}^4$ follows a multivariate normal distribution $N(\mathbf{0}, 4\mathrm{I}_4)$, which makes the noise depend on θ. The basic settings are the same as the quadratic example except that $K = 3000$. Below are the results.



(a) Averaged Relative Error

(b) Asymptotic Proportion

Figure 2: The Quartic Example

	$\Vert\hat{\theta}_K - \theta^*\Vert / \Vert\hat{\theta}_0 - \theta^*\Vert$	p-value versus SU	$Q(\hat{\theta}_K)$
SU	0.1718	NA	0
AVP	0.2703	1.4572×10^{-05}	0.4244
QP	0.2274	0.0045	0.1968
AL	0.1740	0.3878	0.0291

Table 2: Averaged Relative Error, Feasibility and Two-sample Test Result

7 Conclusions

In this paper, we propose the SPSA-based switch updating algorithm for solving constrained stochastic optimization problems. It alternatively updates according to feasibility and conducts one step using the stochastic gradient of the objective loss function estimated by the SPSA algorithm or the gradient of one of the constraint functions. We also show the almost sure convergence result for our algorithm and the asymptotic result for the proportion of iterations that requires loss function measurements. Compared to the SPSA algorithms based on projection or penalty ideas, three dominant advantages exist for the SU algorithm:

(a) Simplicity for implementation and computing;
(b) Feasibility guarantees for final solutions;
(c) No necessity for extra hyper-parameter tuning.

Numerical experiments reveal the efficiency of the SU algorithm over the SPSA algorithm based on penalty ideas and justifies the asymptotic result. It could be postulated that under a large fixed number of loss function measurements, the SU algorithm achieves better performance than the SPSA algorithms based on penalty ideas, with its final solutions closer to the optimal solutions. Together with the advantages above, the SU algorithm is a more efficient and reliable method than the SPSA algorithms based on penalty ideas from an all-round perspective.

Further directions for future research do exist. Initially, switch updating ideas could be applied to more first-order stochastic approximation methods. Additionally, with proper assumptions or algorithmic strategies dealing with the noises added on constraint functions, convergence results could be extended to constrained stochastic optimization problems with noisy constraints (see [9], Chap.5). Finally, more work related to the theoretical results of the convergence rate of the SU algorithm could be conducted.

References

[1] J. C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, 1992.

[2] O. Granichin, V. Erofeeva, Y. Ivanskiy, and Y. Jiang. Simultaneous perturbation stochastic approximation-based consensus for tracking under unknown-but-bounded disturbances. *IEEE Transactions on Automatic Control*, 66(8):3710–3717, 2021.

[3] D. W. Hutchison and S. D. Hill. Simulation optimization of airline delay with constraints. In *Proceeding of the 2001 Winter Simulation Conference (Cat. No. 01CH37304)*, volume 2, pages 1017–1022. IEEE, 2001.

[4] X. Xing and M. Damodaran. Application of simultaneous perturbation stochastic approximation method for aerodynamic shape design optimization. *AIAA journal*, 43(2):284–294, 2005.

[5] Stephen Wright, Jorge Nocedal, et al. Numerical optimization. *Springer Science*, 35(67-68):7, 1999.

[6] P. Sadegh. Constrained optimization via stochastic approximation with a simultaneous perturbation gradient approximation. *IFAC Proceedings Volumes*, 30(11):281–285, 1997.

[7] M. C. Fu and S. D. Hill. Optimization of discrete event systems via simultaneous perturbation stochastic approximation. *IIE Transactions*, 29(3):233–243, March 1997.

[8] J. Shi and J. C. Spall. Sqp-based projection spsa algorithm for stochastic optimization with inequality constraints. In *2021 American Control Conference (ACC)*, pages 1244–1249. IEEE, 2021.

[9] H. J. Kushner and D. S. Clark. *Stochastic approximation methods for constrained and unconstrained systems*, volume 26. Springer Science & Business Media, 2012.

[10] I.-J. Wang and J. C. Spall. Stochastic optimisation with inequality constraints using simultaneous perturbations and penalty functions. *International Journal of Control*, 81(8):1232–1238, August 2008.

[11] Boris Polyak. A general method for solving extremum problems. *Soviet Mathematics. Doklady*, 8, 01 1967.

[12] J. C. Spall. *Introduction to stochastic search and optimization: estimation, simulation, and control*. John Wiley & Sons, 2005.

[13] P. Sadegh and J. C. Spall. Optimal random perturbations for stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 43(10):1480–1484, 1998.

[14] Q. Wang and J. C. Spall. Rate of convergence analysis of discrete simultaneous perturbation stochastic approximation algorithm. In *2013 American Control Conference*, pages 4771–4776. IEEE, 2013.

[15] D. P. Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334, 1997.