Ito and Dhaene *SpringerPlus (2016)5:200*
DOI 10.1186/s40064-016-1823-3

SpringerPlus
a SpringerOpen Journal

METHODOLOGY Open Access

Adaptive initial step size selection for Simultaneous Perturbation Stochastic Approximation

Keiichi Ito[1,2]* and Tom Dhaene[1]

*Correspondence:
itokeiic@gmail.com;
keiichiito@netscape.net
[1] Ghent University - iMinds, INTEC, Gaston Crommenlaan 8 bus 201, Ledeberg, 9050 Ghent, Belgium
Full list of author information is available at the end of the article

Abstract

A difficulty in using Simultaneous Perturbation Stochastics Approximation (SPSA) is its performance sensitivity to the step sizes chosen at the initial stage of the iteration. If the step size is too large, the solution estimate may fail to converge. The proposed adaptive stepping method automatically reduces the initial step size of the SPSA so that reduction of the objective function value occurs more reliably. Ten mathematical functions each with three different noise levels were used to empirically show the effectiveness of the proposed idea. A parameter estimation example of a nonlinear dynamical system is also included.

Keywords: Stochastic approximation, Optimization, Direct method, Noisy function, Parameter estimation

Background

Simultaneous Perturbation Stochastic Approximation (SPSA) (Spall 1992) is an optimization algorithm that uses only objective function measurements in the search of solutions. Applications of SPSA include model-free predictive control (Dong and Chen 2012a, b; Ko et al. 2008), signal timing for vehicle timing control (Spall and Chin 1997), air traffic network (Kleinman et al. 1997), and marine vessel traffic management (Burnett 2004). More applications are mentioned in the introductory article by Spall (1998b). SPSA has been used successfully in many optimization problems that have high-dimensional input parameter space and the objective value is not deterministic (SPSA 2001).

In this optimization method, the initial design parameter vector θ of D-dimensions is perturbed simultaneously in every dimension, i.e. by adding and subtracting a perturbation vector $\mathbf{\Delta}$ of D-dimensions, thus obtaining an estimate of the gradient vector g. Unlike the traditional finite differencing approach, it only takes two function evaluations to obtain the estimate of the gradient. Yet, the number of iteration needed for convergence to the optimum is said to be more or less the same with Finite-Difference Stochastic Approximation (FDSA) (Kiefer and Wolfowitz 1952), which in essence is an approximate steepest-descent method that uses finite-differencing to approximate the partial derivatives along each of the D parameters. Thus, the number of function evaluations of SPSA is D-fold smaller compared to FDSA (Spall 1998b). An extension to this method exists to include second-order (Hessian) effects to accelerate convergence (Spall 2000, 2009; Zhu and Spall 2002). However, we will not treat this enhancement here.

The problem solved by SPSA in this paper can be formulated as following.

$$\min_{\theta \in \Theta} f(\theta), \tag{1}$$

where $f(\theta)$ is the objective function and θ is a D-dimensional vector of parameters. We assume that each element in the vector θ is a real number and has upper and lower bounds that defines the Cartesian product domain Θ. The SPSA and FDSA procedures are in the general recursive form:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}_k(\hat{\theta}_k), \tag{2}$$

where $\hat{g}_k(\hat{\theta}_k)$ is the estimate of the gradient vector $g(\hat{\theta})$ at iteration k based on the measurements of the objective function. The a_k is the step size at iteration k. Equation (2) is analogous to the gradient descent algorithm in nonlinear programming, in which g_k is the gradient of the objective function $\nabla f(\hat{\theta}_k)$. The difference is that in Eq. (2), $\hat{g}_k$ represent gradients stochastically and the effect of the noise or deviation from the true gradient is expected to cancel out as the iteration count k increases. The step sizes a_k are normally prescribed in SPSA and FDSA as a function of k just like the Simulated-Annealing's (Kirkpatrick et al. 1983) cooling schedule. This is because these methods do not assume deterministic responses in the measurements of the objective function values. Thus, unlike the nonlinear programming counterparts, adaptation of step sizes based on gradients and amount of descent achieved (such as in the line search) is usually not done in the stochastic approximation optimization methods. The rationale of the Eq. (2) is intuitively depicted in Fig. 1 for one variable case.

Under appropriate conditions, the iteration in Eq. (2) will converge to the optimum θ^* in some stochastic sense. The hat symbol indicates an "estimate". Thus, $\hat{\theta}_k$ denotes the estimate of the optimum θ^* at iteration k. Let $y(\cdot)$ denote a measurement of the objective

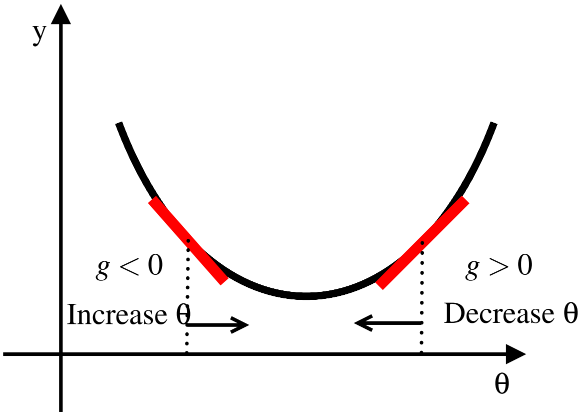

Fig. 1 Objective value minimization using gradient descent (one variable): if gradient g is positive at θ_k then move to $\theta_{k+1} < \theta_k$, if gradient g is negative then move to $\theta_{k+1} > \theta_k$

function $f(\cdot)$ at parameter value denoted by "$\cdot$" and c_k be some small positive number. The measurements are assumed to contain some noise, i.e. $y(\cdot) = f(\cdot) + \text{noise}$. In SPSA, the ith component $\hat{g}_{ki}(\hat{\theta}_k)$ of the gradient vector $\hat{g}_k(\hat{\theta}_k)$ is formed from a ratio involving the individual components in the perturbation vector and the difference in the two corresponding measurements. For two-sided simultaneous perturbations, we have

$$\hat{g}_{ki}(\hat{\theta}_k) = \frac{y(\hat{\theta}_k + c_k \boldsymbol{\Delta}_k) - y(\hat{\theta}_k - c_k \boldsymbol{\Delta}_k)}{2c_k \Delta_{ki}}, \tag{3}$$

where the D-dimensional random perturbation vector

$$\boldsymbol{\Delta}_k = \left(\Delta_{k0}, \Delta_{k1}, \ldots, \Delta_{k(D-1)}\right)^T, \tag{4}$$

follows a specific statistical distribution criterion. Here, i is the parameter index. A simple choice for each component of $\boldsymbol{\Delta}_k$ is to use Bernoulli ±1 distribution, which is essentially a random switching between $+1$ and -1. The Bernoulli distribution is proven to be an optimal distribution for the simultaneous perturbation (Sadegh and Spall 1997). Note also that in the Eq. (3), we do not evaluate $y(\hat{\theta}_k)$. The recursive equation (2) proceeds with only the responses from the two perturbed inputs $y(\hat{\theta}_k + c_k \boldsymbol{\Delta}_k)$ and $y(\hat{\theta}_k - c_k \boldsymbol{\Delta}_k)$.

The choice of a_k and c_k is critical to the performance of SPSA and suggested values can be found in Spall (1998a). At given iteration k:

$$a_k = \frac{a}{(A + k + 1)^{\alpha}}, \tag{5}$$

$$c_k = \frac{c}{(k + 1)^{\gamma}}, \tag{6}$$

where $\alpha = 0.602$, $\gamma = 0.101$, $c \simeq$ standard deviation of measurement noise, $A \leq 10\,\%$ of maximum number of iterations, $a = \delta\hat{\theta}_{0_{\min}} \frac{(A+1)^{\alpha}}{|\hat{g}_{0i}(\hat{\theta}_0)|}$, $k =$ iteration index starting with 0, $\delta\hat{\theta}_{0_{\min}} =$ smallest initial change desired in a parameter.

The setting for α and γ above are not optimal in the asymptotic sense, but are adapted to finite iteration settings. In practice, one of the drawbacks of SPSA is that one has to find good values for a and c, as both affect the performance of the algorithm Spall (2003, pp. 165–166) (Altaf et al. 2011; Shen et al. 2012; Radac et al. 2011; Easterling et al. 2014; Taflanidis and Beck 2008). However, for c, we have a tangible measure, which is the output measurement error (Spall 1998a), to select a proper value up front. If the function response is noiseless, c is usually not a critical parameter. On the other hand, a is more problematic, because no clear measure exists. It is possible to work with $\delta\hat{\theta}_{0_{\min}}$ instead of a, but a priori assignment of its value is still non-trivial if little is known about the function that we are trying to optimize.

A larger value of a generally produces better results compared to a smaller value of a. However, this also increases the chance that the optimization diverges to a worse solution than the starting point. Very often, the user of SPSA has to find as big a as possible that would not cause divergence.

To avoid divergence, an adaptation called "blocking" exists (Easterling et al. 2014; Spall 1998a) in which the objective values at $\hat{\theta}_k$ is evaluated in addition to the two perturbations. If the new objective function value is "significantly worse" than the current objective function value, the updating of $\hat{\theta}_k$ does not happen. The extra function evaluation at each iteration increases the cost of iteration by 33 %. In addition, a problem dependent threshold parameter to block the $\hat{\theta}_k$ update needs to be set up by the user.

Another way to mitigate divergence is to modify the gradient approximation $\hat{g}_k$ by "scaling" and "averaging" (Andradóttir 1996; Xu and Wu 2013). However, the methods proposed in the literature require set up of additional threshold parameters critical to their performance. Furthermore, their methods require additional gradient estimations per iteration.

Stochastic Gradient Descent (SGD) methods use noisy information of the gradient of the objective functions. On the other hand, Stochastic Approximation methods such as FDSA and SPSA only uses measurement of noisy objective values. Therefore, adaptive determination of step sizes based on (approximate) gradients and inverse Hessians in SGD literature (such as in Zeiler (2012), Bottou (2010)) may not be directly applicable to or feasible in SPSA. Convergence conditions also differ between the two. Although this does not exclude the possibility of successful import of ideas from SGD literature, in this paper, we will not delve into this direction.

This paper provides a solution to determine the appropriate values of a by introducing an adaptive scheme as discussed in "Adaptive initial step sizes" section. It does not require any additional objective function evaluations per iteration nor extra problem dependent parameters to set up.

Adaptive initial step sizes

To remedy the sensitivity to a, we propose an adaptive stepping algorithm. At the end of each iteration k, we perform the adjustment described in Algorithm 1.

Algorithm 1 Adaptive Initial Step

```
if $\min\{y(\hat{\theta}_k + c_k \boldsymbol{\Delta}_k), y(\hat{\theta}_k - c_k \boldsymbol{\Delta}_k)\} - y(\hat{\theta}_0) \geq 0$ then
    $\hat{\theta}_{k+1} = \hat{\theta}_b$, where $\hat{\theta}_b$ gives the best $y$ so far
    $a \leftarrow 0.5a$
end if
```

The condition requires that at least one of the two parameter perturbations produce a better (smaller) measurement of the objective function than that of initial guess of parameters $\hat{\theta}_0$ to proceed without modifying a. Therefore, at each iteration k, the smaller of the two measurements of the objective function values of perturbed parameters is compared to that of the initial value at iteration $k = 0$. If the measurements of the objective values of the perturbed parameters are larger, $\hat{\theta}_k$ is reset to θ_b, which is the point that gave the minimum in the history of iteration and a is reduced to half of its previous value. A pseudocode of the proposed SPSA with the adaptive initial step is shown in Algorithm 2. The difference between the standard SPSA and our SPSA is in line 10.

Algorithm 2 Pseudocode of the Proposed Algorithm

```
Initialize $a$ and $c$ (or set $\delta\hat{\theta}_{0_{\min}} \simeq \min(\text{upper bound} - \text{lower bound})$, and $c \simeq$ std of response noise). Set maximum number of iterations maxiter.
Obtain initial measurement $y(\hat{\theta}_0)$, and let $\theta_b = \hat{\theta}_0$.
for k = 0 to maxiter do
    Compute $\mathbf{\Delta}_k$ and $c_k$.
    Evaluate $y(\hat{\theta}_k + c_k\mathbf{\Delta}_k)$ and $y(\hat{\theta}_k - c_k\mathbf{\Delta}_k)$.
    Record the input parameter vector as $\hat{\theta}_b$ if better minimum in $y$ is obtained.
    Compute $\hat{g}_{ki}(\hat{\theta}_k)$.
    Compute $a_k$.
    $\hat{\theta}_{k+1} = \hat{\theta}_k - a_k\hat{g}_k(\hat{\theta}_k)$.
    Perform Algorithm 1.
end for
```

Comments on convergence

Currently available theories of stochastic algorithms are almost all based on asymptotic properties with $k \to \infty$, and SPSA is no exception. For given conditions Spall (2003, p. 183), SPSA is proven to converge to a local optima almost surely. However, under limited function evaluation budget, we frequently encounter situations in which SPSA returns worse solution than the initial i.e. divergence. The method we propose is a practical remedy conceived in a finite k setting. We will show, in the next section, its effectiveness empirically via numerical experiments with k in the order of 10^3.

For $\hat{\theta}_k$ to converge to the optimal solution θ^* in *infinite* steps, the following conditions are required for a_k and c_k (Spall 1992): $a_k, c_k > 0$ for all k; $a_k, c_k \to 0$ as $k \to \infty$; $\sum_{k=0}^{\infty} a_k = \infty$, and $\sum_{k=0}^{\infty} \left(\frac{a_k}{c_k}\right)^2 < \infty$. With Algorithm 1, $\sum_{k=0}^{\infty} a_k = \infty$ is not guaranteed. For example, if the reduction of a happens in every iteration k, the sum is convergent. In practice, the numbers of function evaluations are finite, and reductions of a are expected to happen only a limited number of times. Therefore, this violation is expected to pose little problem.

The intention of the proposed method is not to modify the asymptotic convergence rate of the original SPSA algorithm Spall (2003, pp. 186–188). The adaptive step takes place only if it is suspected that the objective value has become larger than at the starting point $\hat{\theta}_0$. The probability of Algorithm 1 taking place is expected to go to zero under reasonable signal-to-noise ratio as $f(\hat{\theta}_k)$ decreases. The worst situation that can happen is that the every perturbation $c_k\mathbf{\Delta}_k$ produces worsening moves and no improvement is obtained compared to the starting point θ_0. In "Computational results" section, we will confirm empirically what we have described about the convergence in finite k settings ($k \sim 10^3$).

Another reason to take the objective value at the starting point as the threshold value to judge divergence is that if we update this value with $y(\hat{\theta}_k)$, where $k > 0$, we may risk picking a point that is too low due to the noise incurred in the measurement y. This in turn inhibits further improvement of $\hat{\theta}_k$ for lower objective values.

In the following section, the smallest output of mathematical functions will be sought using the standard SPSA and our adaptive initial stepping SPSA. This will show the sensitivity of the function value in the final iteration to the initial step size $\delta\hat{\theta}_{0_{\min}}$ and so the sensitivity to a, and how the adaptive initial stepping substantially mitigates the difficulty to find the proper initial perturbation magnitude.

Computational results

In this section, we will compare the original SPSA and our modified SPSA as described in Algorithm 2 using 10 analytical test functions and a parameter estimation example of a nonlinear dynamic system.

Test functions

To see the effect of the new adaptive stepping algorithm in SPSA, the minimum points of ten different mathematical test functions were sought. Except for Griewank function, the following conditions were applied. The functions' responses were minimized from arbitrary starting points $\hat{\theta}_0 \in [-2, 2]^D$ (D-dimensional product space with lower bound -2 and upper bound 2). If $\hat{\theta}_k = [\hat{\theta}_{k0}, \hat{\theta}_{k1}, \cdots, \hat{\theta}_{ki}, \cdots, \hat{\theta}_{k(D-1)}]^T$ exceeded $[-10,\ 10]$ in any of its D dimensions, that parameter was replaced by -10 if it was less than -10 or was replaced by 10 if it was larger than 10. For Griewank function, it was randomly started from $\hat{\theta}_0 \in [-120,\ 120]^D$. If $\hat{\theta}_k$ exceeded $[-600, 600]$ in any of its D dimensions, that parameter was replaced by -600 if it was less than -600 or was replaced by 600 if it was larger than 600. For all ten functions, the iteration was stopped when 2000 evaluations of the objective function were reached. For convenience, we will label our proposed algorithm as "A_SPSA" and the standard SPSA as "SPSA".

The optimizations for each of the ten objective functions were started from 20 different starting points. After the 2000 iterations, the distributions of objective values were plotted with respect to $\delta\hat{\theta}_{0_{\min}}$. Eleven different values of $\delta\hat{\theta}_{0_{\min}}$ between 1.0×10^{-4} and 1.0×10^{1} (up to 1.0×10^{2} for Griewank) were used to make the plot. The dimensions of the functions were set to be 20, i.e. $D = 20$.

The definitions of the ten functions are given in the following. The Rosenbrock function is described as

$$
\begin{aligned}
f(\theta) &= \sum_{i=0}^{D-2} \left(100(\theta_{i+1} - \theta_i^2)^2 + (\theta_i - 1)^2\right), \\
&\quad i = 0, 1, \ldots, D-1, \quad D > 1, \\
&\quad f(\theta^*) = 0, \quad \theta_i^* = 1.
\end{aligned}
\tag{7}
$$

The Sphere function is described as

$$
\begin{aligned}
f(\theta) &= \sum_{i=0}^{D-1} \theta_i^2, \\
&\quad i = 0, 1, \ldots, D-1, \\
&\quad f(\theta^*) = 0, \quad \theta_i^* = 0.
\end{aligned}
\tag{8}
$$

The Schwefel function is described as

$$
\begin{aligned}
f(\theta) &= \sum_{j=0}^{D-1} \left(\sum_{i=0}^{j} \theta_i\right)^2, \\
&\quad i = 0, 1, \ldots, D-1, \\
&\quad f(\theta^*) = 0, \quad \theta_i^* = 0.
\end{aligned}
\tag{9}
$$

The Rastrigin function is described as

$$\begin{aligned} f(\theta) &= \sum_{i=0}^{D-1} \left(\theta_i^2 - 10\cos(2\pi\theta_i) + 10\right), \\ i &= 0, 1, \ldots, D-1, \\ f(\theta^*) &= 0, \quad \theta_i^* = 0. \end{aligned} \tag{10}$$

The Skewed Quartic function Spall (2003, ex. 6.6) is described as

$$\begin{aligned} f(\theta) &= (\mathbf{B}\theta)^T\mathbf{B}\theta + 0.1\sum_{i=0}^{D-1}(\mathbf{B}\theta)_i^3 + 0.01\sum_{i=0}^{D-1}(\mathbf{B}\theta)_i^4, \\ i &= 0, 1, \ldots, D-1, \\ f(\theta^*) &= 0, \quad \theta_i^* = 0. \end{aligned} \tag{11}$$

where the matrix **B** in the Skewed Quartic function is a square matrix with upper triangular elements set to 1 and the lower triangular elements set to zero. The Griewank function is described as

$$\begin{aligned} f(\theta) &= 1 + \sum_{i=0}^{D-1}\frac{\theta_i^2}{4000} - \prod_{i=0}^{D-1}\cos(\frac{\theta_i}{\sqrt{i}}), \\ i &= 0, 1, \ldots, D-1, \\ f(\theta^*) &= 0, \quad \theta_i^* = 0. \end{aligned} \tag{12}$$

The Ackley function is described as

$$\begin{aligned} f(\theta) = &-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=0}^{D-1}\theta_i^2}\right) \\ &- \exp\left(\frac{1}{D}\sum_{i=0}^{D-1}\cos(2\pi\theta_i)\right) \\ &+ 20 - \exp(1), \\ &\; i = 0, 1, \ldots, D-1, \\ &\; f(\theta^*) = 0, \quad \theta_i^* = 0. \end{aligned} \tag{13}$$

The Manevich function is described as

$$\begin{aligned} f(\theta) &= \sum_{i=0}^{D-1}\left[(1-\theta_i)^2/2^j\right], \\ i &= 0, 1, \ldots, D-1, \\ f(\theta^*) &= 0, \quad \theta_i^* = 1. \end{aligned} \tag{14}$$

The Ellipsoid function is described as

$$\begin{aligned} f(\theta) &= \sum_{i=0}^{D-1} i\theta_i^2, \\ i &= 0, 1, \ldots, D-1, \\ f(\theta^*) &= 0, \quad \theta_i^* = 0. \end{aligned} \tag{15}$$

The Rotated Ellipsoid function is described as

$$\begin{aligned} f(\theta) &= \sum_{i=0}^{D-1}\left(\sum_{j=0}^{i}\theta_j^2\right)^2, \\ i &= 0,1,\ldots,D-1, \\ f(\theta^*) &= 0, \quad \theta_i^* = 0. \end{aligned} \tag{16}$$

Each of Figs. 2, 3, 4, 5, 6, 7, 8, 9, 10 and 11 show three different cases of noisy measurements of the outputs. The subfigures (a) have no noise added, subfigures (b) and (c) have Gaussian noise added to the true output with standard deviation σ of 0.1 and 1.0 respectively. In all the three noise levels of the ten functions, $c = 0.2$ was used.

A general trend observed from the figures is that when the initial step size is large, the original SPSA tends to diverge to big objective values. The SPSA with the proposed initial step size reduction, on the other hand, effectively mitigates this divergence problem producing smaller objective values in general as the (a priori) initial step size is increased. This is because if the two function evaluations in the iteration are not smaller than the starting point value $f(\hat{\theta_0})$, the algorithm will reduce the step size (by halving a) and restart at $\hat{\theta_b}$, which is the point that gave the smallest output in the history of iterations. However, note that the iteration index k in a_k and c_k is not reinitialized. For the ten functions tested, A_SPSA achieved its best performance when $\delta\hat{\theta}_{0_{\min}}$ was close to 10 or 100 for Griewank function. This indicates that one can simply set the minimum perturbation $\delta\hat{\theta}_{0_{\min}}$ close to the magnitude of the difference between upper and lower bound

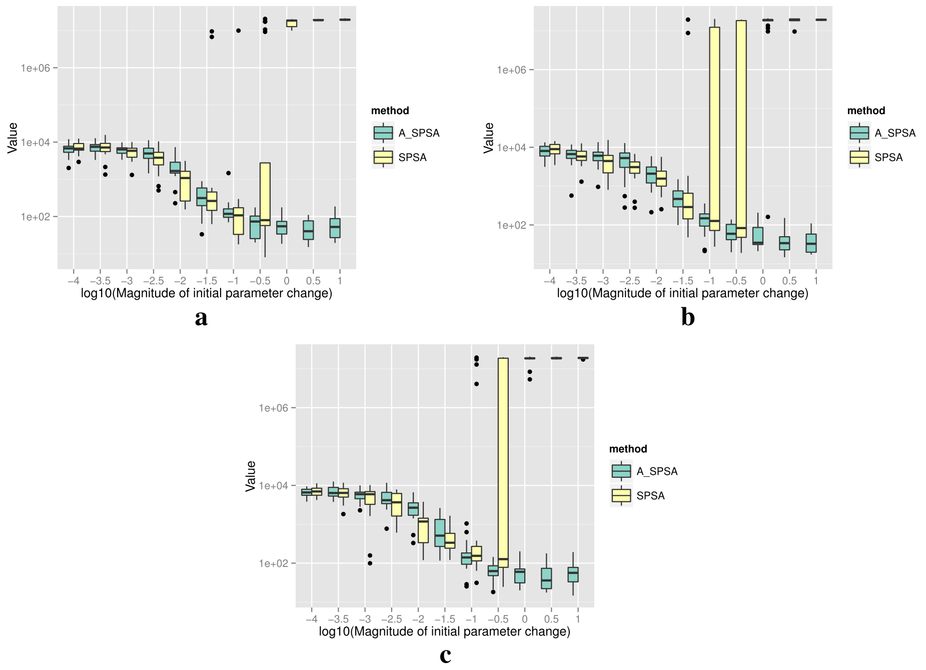

Fig. 2 Initial parameter change $\delta\hat{\theta}_{0_{\min}}$ and distribution of responses after 2000 function evaluations for "Rosenbrock". **a** No noise, **b** σ = 0.10, **c** σ = 1.0

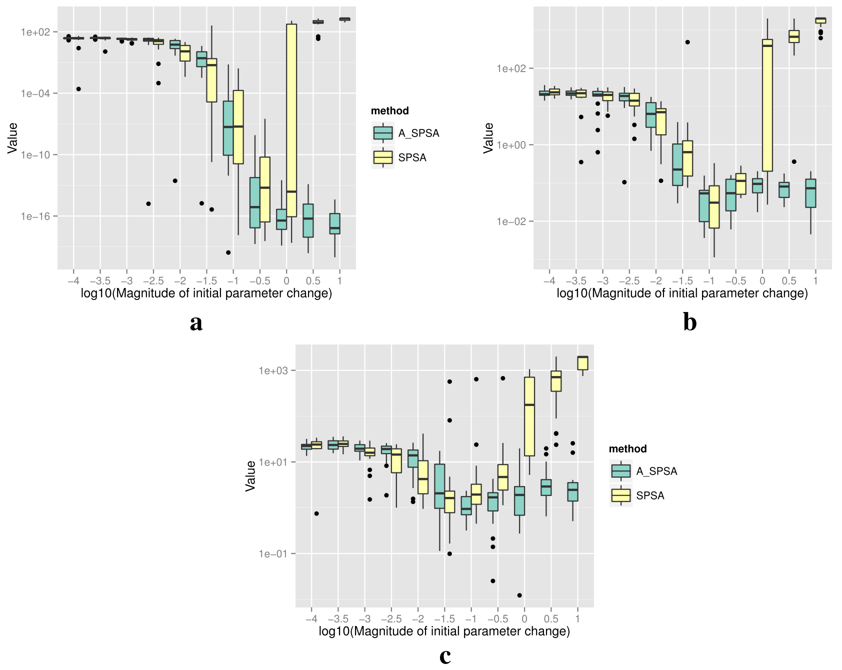

Fig. 3 Initial parameter change $\delta\hat{\theta}_{0_{\min}}$ and distribution of responses after 2000 function evaluations for "Sphere". **a** No noise, **b** σ = 0.10, **c** σ = 1.0

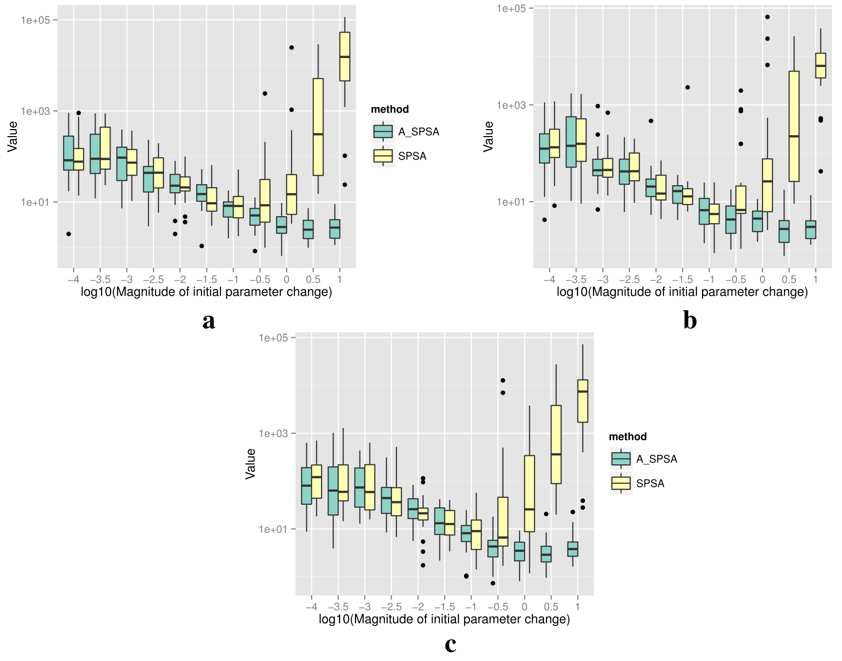

Fig. 4 Initial parameter change $\delta\hat{\theta}_{0_{\min}}$ and distribution of responses after 2000 function evaluations for "Schwefel". **a** No noise, **b** σ = 0.10, **c** σ = 1.0

of the parameter in consideration. This may not be a guarantee for the best results but doing so does not cause the optimization to diverge to large responses and the results achieved are not substantially worse than the cases with best settings for *a*.

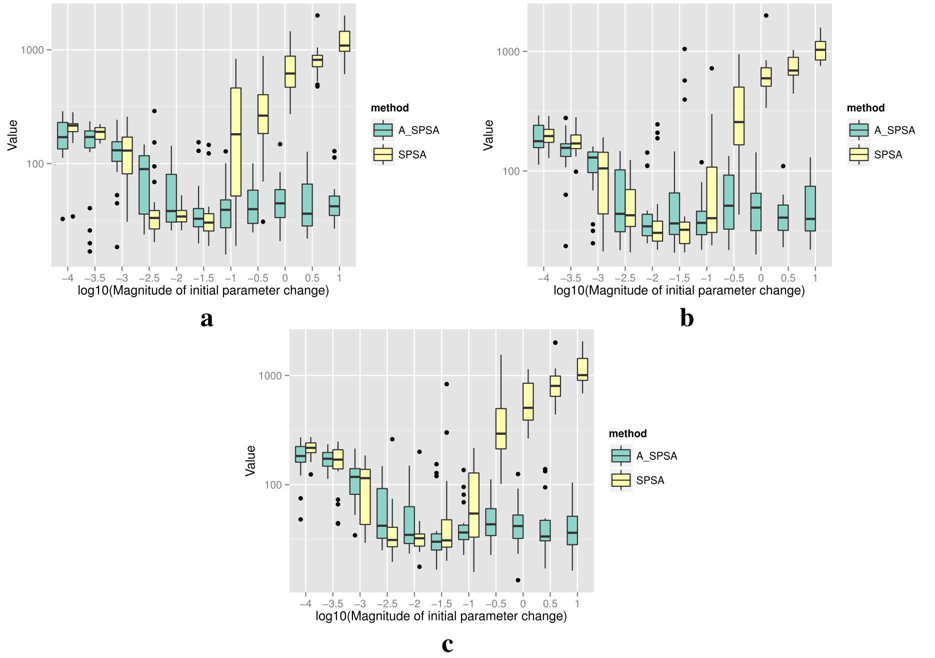

Fig. 5 Initial parameter change $\delta\hat{\theta}_{0_{\min}}$ and distribution of responses after 2000 function evaluations for "Rastrigin". **a** No noise, **b** σ = 0.10, **c** σ = 1.0

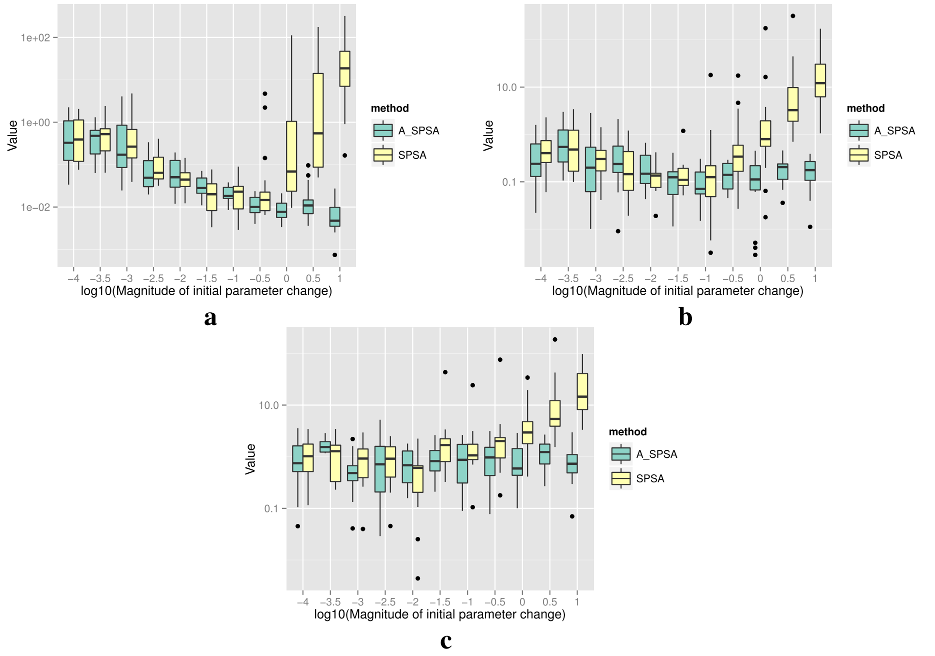

Fig. 6 Initial parameter change $\delta\hat{\theta}_{0_{\min}}$ and distribution of responses after 2000 function evaluations for "Skewed Quartic". **a** No noise, **b** σ = 0.10, **c** σ = 1.0

As mentioned earlier, the value for c is important when the measurements of y contain noise. Figure 12 shows how the choice of c affects the outcome of optimizations. The figure shows the case of the 20 dimensional Sphere Function with Gaussian noise

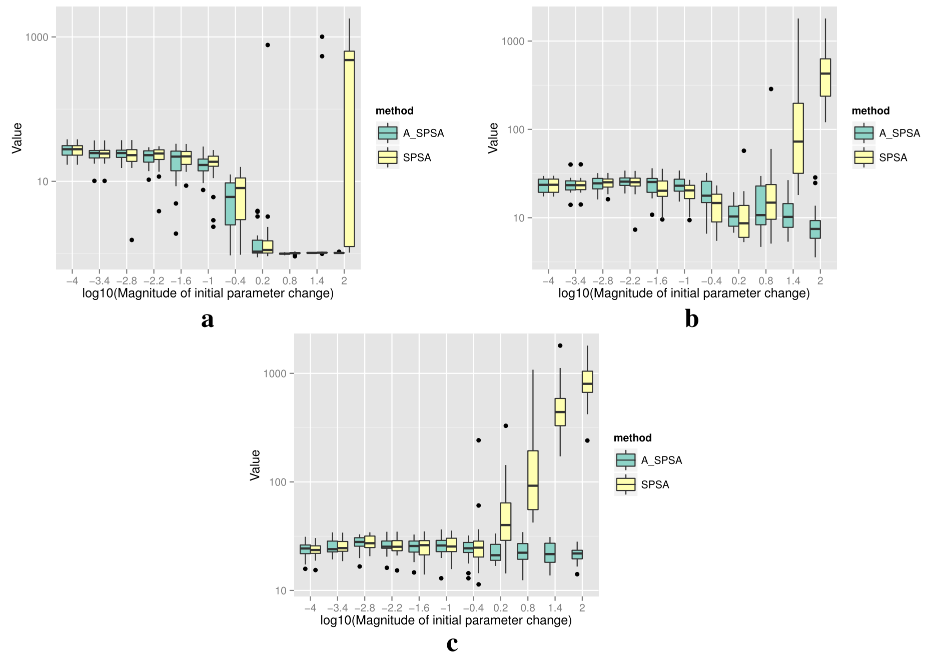

Fig. 7 Initial parameter change $\delta\hat{\theta}_{0_{\min}}$ and distribution of responses after 2000 function evaluations for "Griewank". **a** No noise, **b** σ = 0.10, **c** σ = 1.0

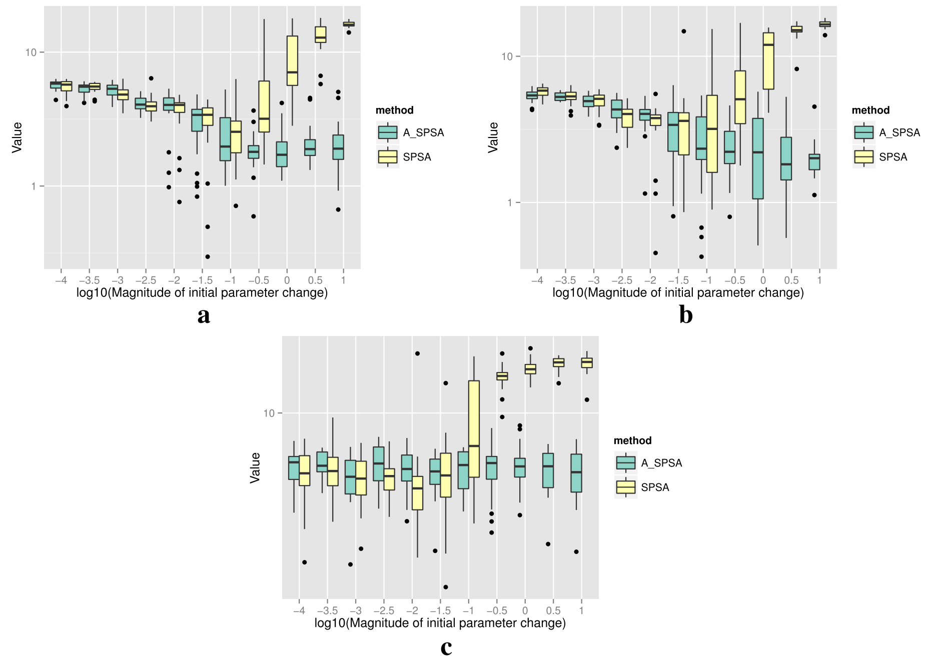

Fig. 8 Initial parameter change $\delta\hat{\theta}_{0_{\min}}$ and distribution of responses after 2000 function evaluations for "Ackley". **a** No noise, **b** σ = 0.10, **c** σ = 1.0

having standard deviation $\sigma = 0.1$. Among the three values of c, namely 0.01, 0.1 and 1.0, $c = \sigma = 0.1$ gave the best results for A_SPSA. At $c = 1.0$, however, A_SPSA showed little improvement in the objective value regardless of $\delta\hat{\theta}_{0_{\min}}$ magnitude. This is caused

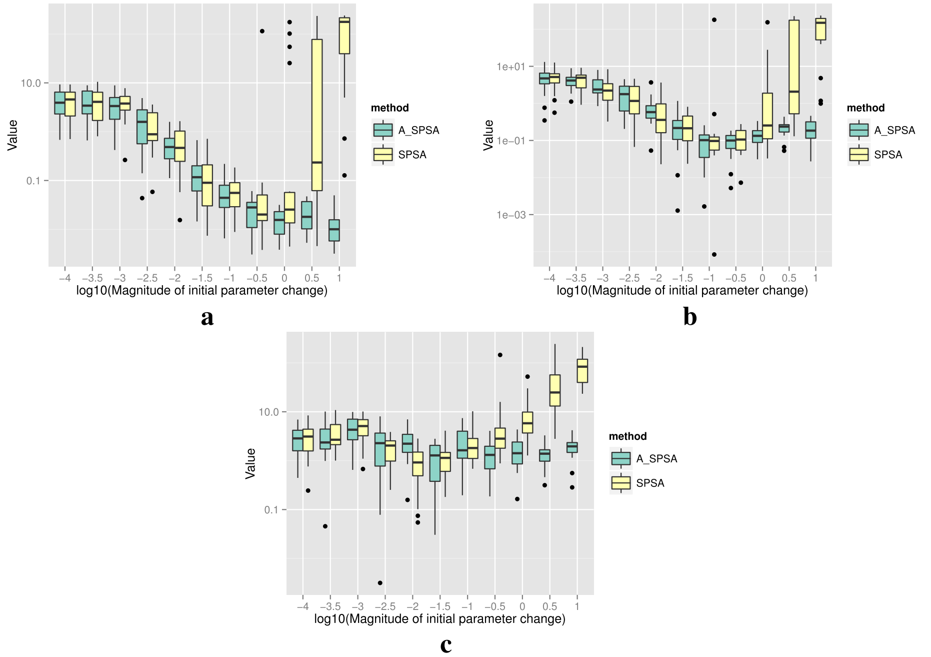

Fig. 9 Initial parameter change $\delta\hat{\theta}_{0_{\min}}$ and distribution of responses after 2000 function evaluations for "Manevich". **a** No noise, **b** σ = 0.10, **c** σ = 1.0

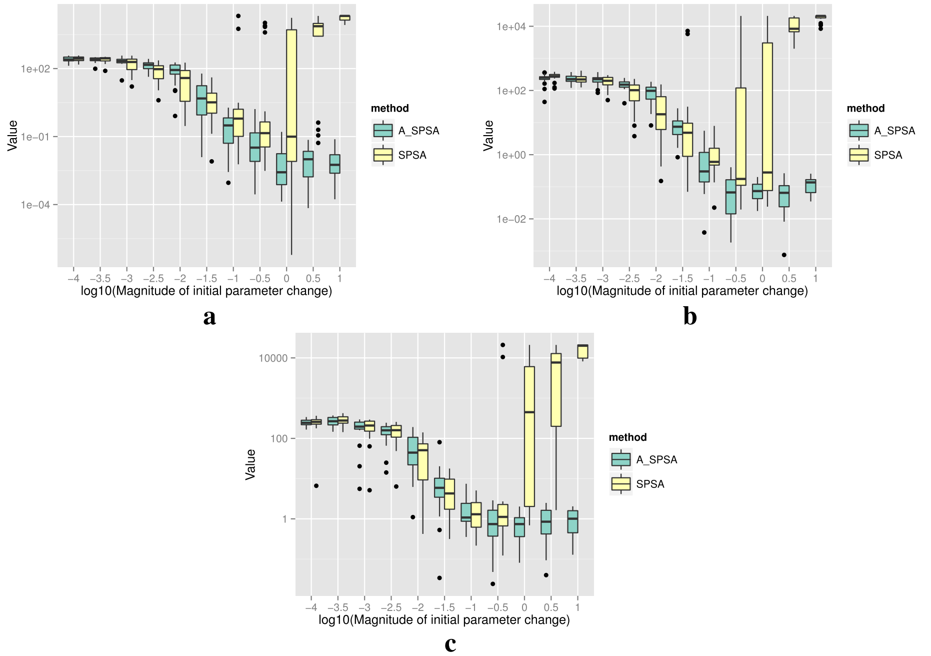

Fig. 10 Initial parameter change $\delta\hat{\theta}_{0_{\min}}$ and distribution of responses after 2000 function evaluations for "Ellipsoid". **a** No noise, **b** σ = 0.10, **c** σ = 1.0

by *a* becoming prematurely too small in the divergent early iterations. On the other hand, the standard SPSA showed a good reduction at $\log_{10}(\delta\hat{\theta}_{0_{\min}}) = -2.0$, and −1.5. at both $c = 0.1$ and 1.0. This implies that for A_SPSA, a range of values of good *c* can be

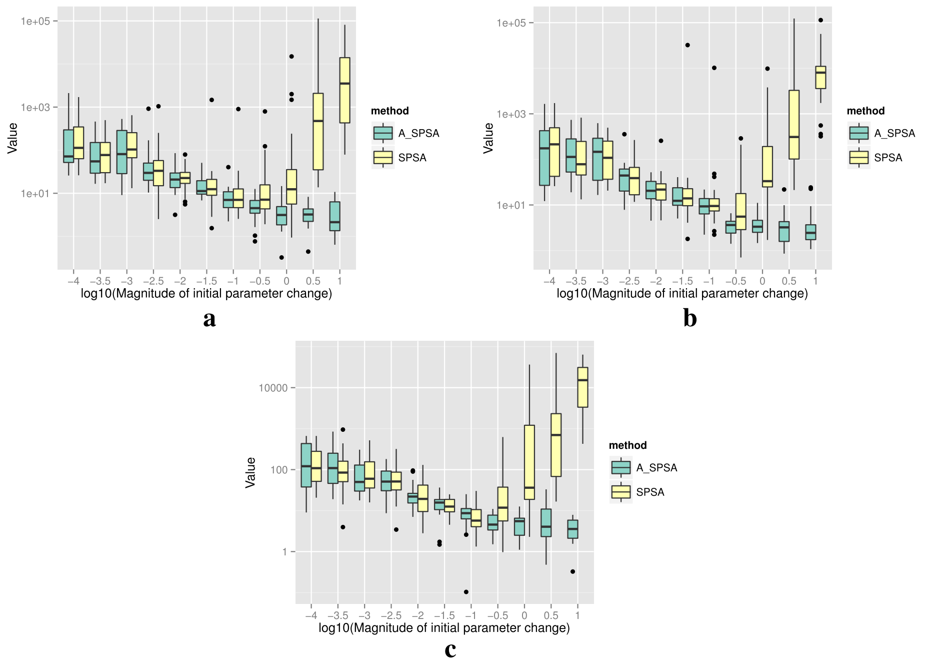

Fig. 11 Initial parameter change $\delta\hat{\theta}_{0_{\min}}$ and distribution of responses after 2000 function evaluations for "Rotated Ellipsoid". **a** No noise, **b** σ = 0.10, **c** σ = 1.0

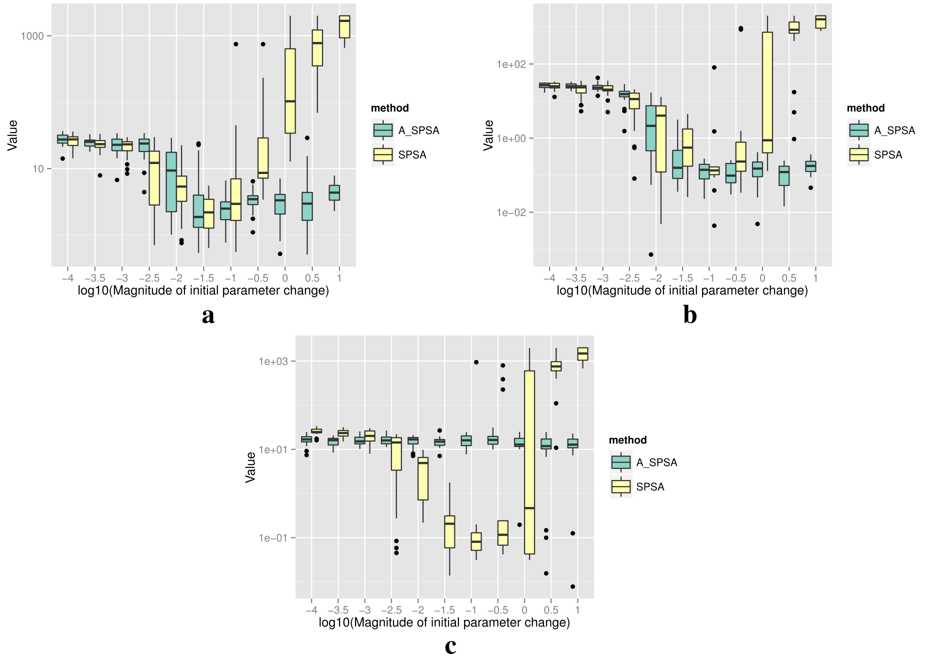

Fig. 12 Effect of choice of *c* to the final response of "Sphere" with Gaussian noise of $\sigma\ = 0.1$after 2000 function evaluations. **a** $c = 0.01$, **b** $c = 0.10$, **c** $c = 1.00$

narrower than that of the standard SPSA. On the other hand, the choice of $\delta\hat{\theta}_{0_{\min}}$ (and therefore a) is much easier for A_SPSA. We can, for example, let $\delta\hat{\theta}_{0_{\min}} \simeq \min(U-L)$, where $\min(U-L)$ is the minimum difference between upper and lower bounds of the domain of parameter vector θ. In practice, it is better to scale all the input dimensions to fall in similar or equal intervals.

Figure 13 shows the results of optimizing the Rosenbrock and Rastrigin functions using three different values of multiplication factor of a: 0.1, 0.5, and 0.9. The difference in multiplication factor does not change the general trend that larger $\delta\hat{\theta}_{0_{\min}}$ produces better results and that divergence does not occur. One could tune the value of the multiplication factor, but the default value of 0.5 that we showed in the Algorithm 1 generally produces satisfactory results compared to other values of multiplication factors between 0 and 1. The Fig. 13 (b) also shows that $\delta\hat{\theta}_{0_{\min}} \simeq \min(U-L)$ may not be an optimal setting since smaller value $\delta\hat{\theta}_{0_{\min}} \simeq 10^{-1.5}$ is shown to produce better optimization results when the reduction rate is slow at 0.9. This implies that in a bumpy (highly multimodal) function like Rastrigin, the slow decrease in a can adversely affect the minimization of the objective value by a large number of resets to θ_b. The opposite is true with Rosenbrock function in (a), in which the slow reduction factor 0.9 gave the best result at $\delta\hat{\theta}_{0_{\min}} \simeq 10^1$.

For all the mathematical functions tested in this paper, optimization using SPSA diverges almost surely if the $\delta\hat{\theta}_{0_{\min}}$ is large. However, A_SPSA and SPSA give closely matching results when the initial step sizes are relatively small (i.e., the left hand side of the plots in Figs. 2, 3, 4, 5, 6, 7, 8, 9, 10 and 11). This is because, in cases that divergence does not happen, the adaptation of a does not take place in A_SPSA and therefore SPSA and A_SPSA have identical behavior. This is a confirmation that Algorithm 1 does not alter, in any significant way, the finite sample convergence characteristics of the original SPSA when the divergence does not manifest.

Nonlinear dynamics example

We consider a parameter estimation problem with Lorenz attractor. Its nonlinear dynamics is described as

$$\frac{dx_1}{dt} = s(x_2 - x_1), \tag{17}$$

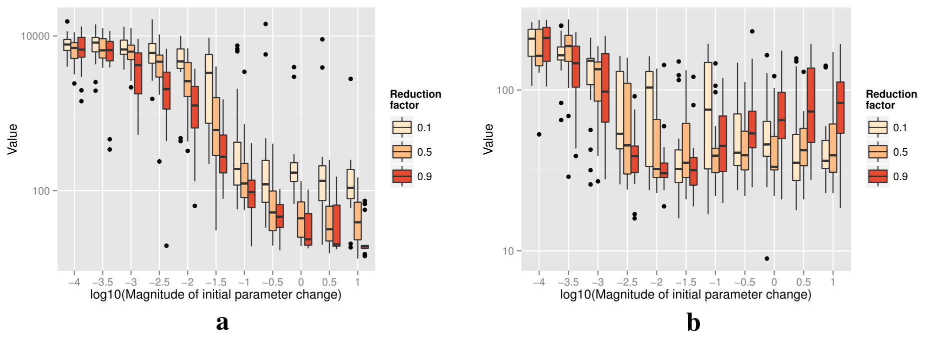

Fig. 13 Effect of choice of the reduction factor of *a* to the responses after 2000 function evaluations. **a** Rosenbrock (no noise), **b** Rastrigin (no noise)

$$\frac{dx_2}{dt} = x_1(r - x_3) - x_2, \tag{18}$$

$$\frac{dx_3}{dt} = x_1 x_2 - b x_1. \tag{19}$$

We seek to identify the system parameters $\theta = [s, r, b]$ by minimizing the one-time-step-ahead prediction error L_k of the state $\mathbf{x}_{k+1}$ given the current state $\mathbf{x}_k = [x_{k1}, x_{k2}, x_{k3}]^T$. We use fourth-order Runge–Kutta method to obtain $\mathbf{x}_{k+1}$.

Let us denote $\hat{\mathbf{x}}_{k+1}$ as one-time-step-ahead prediction given by the estimated system with parameters $\hat{\theta}_k$. Then, we can define the prediction error as

$$L_k(\mathbf{x}_k, \hat{\theta}_k) = [\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}]^T \cdot [\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}]. \tag{20}$$

Thus, the optimization to be solved is

$$\min_{\theta \in \Theta} L_k(\mathbf{x}_k, \theta). \tag{21}$$

The index k above is the same as the index k in the SPSA algorithms. So the SPSA iteration proceeds along with the time steps of the dynamic system to compute L_k.

We set the true parameters to be $\theta = [10, 28, 8/3]$ and pretend to not to know them. We set the time increment to be $\Delta t = 0.005$ and simulate from $t = 0$ to 20, obtaining target state $\mathbf{x}_k$ with $k = 0, 1, 2, \ldots, 4000$. We let $\delta\hat{\theta}_{0_{\min}} \in \{0.001, 0.01, 1, 10, 100, 1000\}$ and at each value of $\delta\hat{\theta}_{0_{\min}}$ we run both A_SPSA and SPSA 20 times.

For this problem, we set the parameter space as three-dimensional product space $\boldsymbol{\Theta} = [0, 500]^3$. The initial state is $\mathbf{x}_0 = [2, 3, 4]^T$. The initial guess (starting point) of the parameter set $\hat{\theta}_0$ is a random pick from $\boldsymbol{\Theta}$.

Figure 14 show the box plots of final L_k when started from different values of $\delta\hat{\theta}_{0_{\min}}$. The smallest median of final L_k is obtained at $\delta\hat{\theta}_{0_{\min}} = 10$ for SPSA and $\delta\hat{\theta}_{0_{\min}} = 100$ and 1000 for A_SPSA. The best medians of final L_k obtained for A_SPSA (5.62×10^{-15}) is

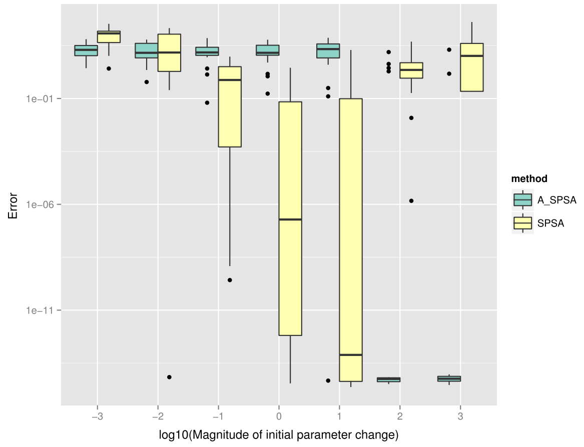

Fig. 14 Initial parameter change $\delta\hat{\theta}_{0_{\min}}$ and distribution of L_{4000} (after 8000 function evaluations)

smaller compared to that of SPSA (3.10×10^{-13}). However, both SPSA and A_SPSA had some runs that did not converge to the above mentioned near-zero L_k values even at these $\delta\hat{\theta}_{0_{\min}}$.

Again, for A_SPSA, the best setting were obtained when $\delta\hat{\theta}_{0_{\min}}$ was set to large values near the order of magnitude of the distance between upper and lower bound of the domain, while for SPSA, the best $\delta\hat{\theta}_{0_{\min}}$ was at an interior value between 10^{-3} and 10^{3}.

Figure 15 shows the trajectory of the reference Lorenz attractor and the simulation of the Lorenz attractor whose system parameters s, r, and b were successfully identified by A_SPSA. The time t is run from 0 to 20 starting from the same initial condition used in the identification. The figure shows excellent match.

Figure 16 shows the box plots of parameters estimated by A_SPSA and SPSA starting at their best $\delta\hat{\theta}_{0_{\min}}$ settings. The corresponding statistics are shown in Tables 1 and 2. The boxes appear collapsed as single horizontal lines at medians since the spaces between

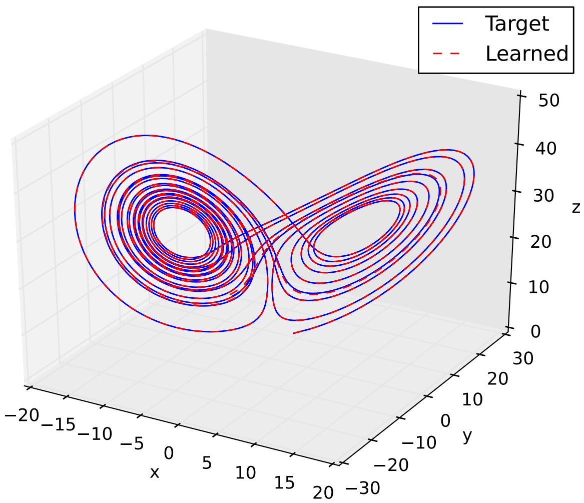

Fig. 15 State evolution of the target and identified Lorenz attractor, $t = 0$ to 20

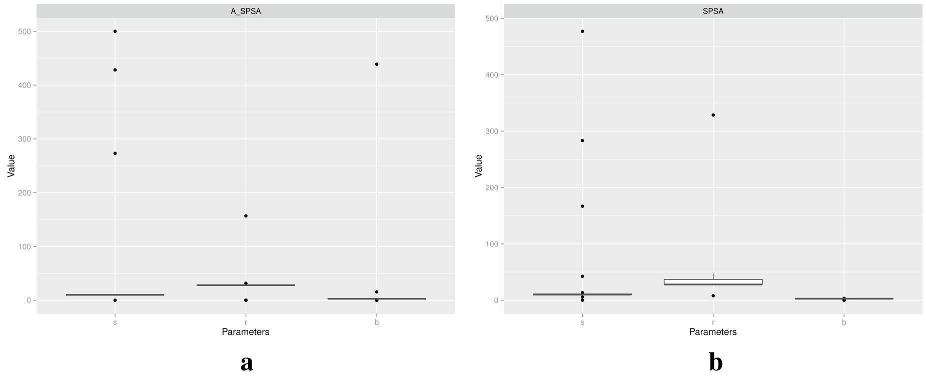

Fig. 16 Distribution of the parameters identified by A_SPSA and SPSA. **a** A_SPSA with $\delta\hat{\theta}_{0_{\min}} = 100$, **b** SPSA with $\delta\hat{\theta}_{0_{\min}} = 10$

Table 1 Statistics of identified Lorenz Attractor parameters by 20 SPSA runs at $\delta\hat{\theta}_{0_{\min}} = 10$

	Method	*s*	*r*	*b*	Pred. Err. L_{4000}
1	A_SPSA: 0	Min.: 0.00	Min.: 8.017	Min.: 0.000	Min.: 0.0000
2	SPSA: 20	1st Qu.: 10.00	1st Qu.: 28.000	1st Qu.: 2.642	1st Qu.: 0.0000
3		Median: 10.00	Median: 28.000	Median: 2.667	Median: 0.0000
4		Mean: 55.94	Mean: 45.534	Mean: 2.311	Mean: 1.3645
5		3rd Qu.: 11.11	3rd Qu.: 36.817	3rd Qu.: 2.667	3rd Qu.: 0.1017
6		Max.: 477.04	Max.: 328.504	Max.: 3.261	Max.: 19.6773

Table 2 Statistics of identified Lorenz attractor parameters by 20 A_SPSA runs at $\delta\hat{\theta}_{0_{\min}} = 100$

	Method	*s*	*r*	*b*	Pred. Err. L_{4000}
1	A_SPSA: 20	Min.: 0.000	Min.: 0.000	Min.: 0.0000	Min.: 0.0000
2	SPSA: 0	1st Qu.: 10.000	1st Qu.: 28.000	1st Qu.: 2.6667	1st Qu.: 0.0000
3		Median: 10.000	Median: 28.000	Median: 2.6667	Median: 0.0000
4		Mean: 68.069	Mean: 31.816	Mean: 24.8487	Mean: 1.2328
5		3rd Qu.: 10.000	3rd Qu.: 28.000	3rd Qu.: 2.6667	3rd Qu.: 0.0000
6		Max.: 500.000	Max.: 156.811	Max.: 438.8246	Max.: 15.6654

first quartiles and third quartiles are very narrow. Some non-converging cases are visible as dots on the figure. The figure and the tables show that the parameter estimates are more consistent from run to run in A_SPSA than that of SPSA as A_SPSA has narrower first and third quartile differences.

Conclusion

With the adaptive initial step algorithm, one can avoid divergence in SPSA iterations. Moreover, with a large initial step size, the SPSA algorithm with the adaptive initial step algorithm was able to find equal or better solutions compared to the original SPSA for all the ten mathematical function minimization problems that we have tested. In the nonlinear dynamics example, the new algorithm was able to find system parameters more precisely. The proposed method may not eliminate the need of tuning the parameters of SPSA algorithms, but it facilitates the process by eliminating the risk of solution divergence and reducing the trial-and-error effort. Further testing of the algorithm with different test functions, noise distributions, and industrial use-cases would be beneficial. The improvement proposed in this paper is expected to be valuable when the objective functions are costly to evaluate or if the algorithm is employed inside another algorithm such as machine learning or target tracking, for manual tuning of the parameters would be cumbersome in such cases. As a future work, it would be beneficial to investigate under what conditions the probability of the proposed adaptation (i.e. going into if-branch in Algorithm 1) happening tends to zero as iteration *k* tends to infinity.

Authors' contributions

KI has conceived the Algorithm 1, conducted the numerical experiments, and has written this manuscript. TD has provided guidance by supervising the progress of the research, and has reviewed the manuscript for intellectual feedback and editing.

Author details
[1] Ghent University - iMinds, INTEC, Gaston Crommenlaan 8 bus 201, Ledeberg, 9050 Ghent, Belgium. [2] Noesis Solutions, Gaston Geenslaan 11 B4, 3001 Louvain, Belgium.

Acknowledgements
The authors would like to thank James C. Spall for constructive comments on the proposed method. Keiichi Ito has been funded by the Institute for the Promotion of Innovation through Science and Technology (IWT) through the Baekeland Mandate program. This research has also been funded by the Interuniversity Attraction Poles Programme BESTCOM initiated by the Belgian Science Policy Office.

Competing interests
Keiichi Ito is employed at Noesis Solutions, a private company, which has interests in optimization methods.

Received: 13 September 2015 Accepted: 15 February 2016
Published online: 27 February 2016

References

Altaf MU, Heemink AW, Verlaan M, Hoteit I (2011) Simultaneous perturbation stochastic approximation for tidal models. Ocean Dyn 61:1093–1105

Andradóttir S (1996) A scaled stochastic approximation algorithm. Manag Sci 42(4):475–498. doi:10.1287/mnsc.42.4.475

Bottou L (2010) Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT'2010. Springer, pp 177–186

Burnett R (2004) Application of stochastic optimization to collision avoidance. In: Proceedings of the American control conference. Massachusetts, USA, Boston, pp 2789–2794

Dong N, Chen Z (2012) A novel ADP based model-free predictive control. Nonlinear Dyn 69(1–2):89–97. doi:10.1007/s11071-011-0248-3

Dong N, Chen Z (2012) A novel data based control method based upon neural network and simultaneous perturbation stochastic approximation. Nonlinear Dyn 67(2):957–963. doi:10.1007/s11071-011-0039-x

Easterling D, Watson L, Madigan M, Castle B, Trosset M (2014) Parallel deterministic and stochastic global minimization of functions with very many minima. Comput Optim Appl 57(2):469–492. doi:10.1007/s10589-013-9592-1

Kiefer J, Wolfowitz J (1952) Stochastic estimation of the maximum of a regression function. Ann Math Stat 23:452–466

Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680

Kleinman NL, Hill SD, Ilenda VA (1997) SPSA/SIMMOD optimization of air traffic delay cost. In: Proceedings of the American control conference. Albuquerque, New Mexico, USA, pp 1121–1125

Ko HS, Lee KY, Kim HC (2008) A simultaneous perturbation stochastic approximation (SPSA)-based model approximation and its application for power system stabilizers. Int J Control Autom Syst 6(4):506–514

Radac M, Precup R, Petriu E, Preitl S (2011) Application of IFT and SPSA to servo system control. IEEE Trans Neural Netw 22(12):2363–2375. doi:10.1109/TNN.2011.2173804

Sadegh P, Spall JC (1997) Optimal random perturbations for stochastic approximation using a simultaneous perturbation gradient approximation. In: Proceedings of the American control conference. Albuquerque, NM, USA, pp 3582–3586

Shen X, Yao M, Jia W, Yuan D (2012) Adaptive complementary filter using fuzzy logic and simultaneous perturbation stochastic approximation algorithm. Measurement 45(5):1257–1265. doi:10.1016/j.measurement.2012.01.011. http://www.sciencedirect.com/science/article/pii/S0263224112000267

Spall JC (1992) Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE Trans Autom Control 37(3):332–341

Spall JC (1998) Implementation of the simultaneous perturbation algorithm for stochastic optimization. IEEE Trans Aerosp Electron Syst 34(3):817–823

Spall JC (1998) An overview of the simultaneous perturbation method for efficient optimization. Johns Hopkins APL Tech Dig 19(4):482–492

Spall JC (2000) Adaptive stochastic approximation by the simultaneous perturbation method. Trans Autom Control 45(10):1839–1853

Spall JC (2003) Introduction to stochastic search and optimization, estimation, simulation and control. Wiley-Interscience, New York

Spall JC (2009) Feedback and weighting mechanisms for improving jacobian estimates in the adaptive simultaneous perturbation algorithm. IEEE Trans Autom Control 54(6):1216–1229

Spall JC, Chin DC (1997) Traffic-responsive signal timing for system-wide traffic control. Transp Res 5(Part C):153–163

SPSA (2001) Simultaneous perturbation stochastic approximation: a method for system optimization. http://www.jhuapl.edu/SPSA/index.html

Taflanidis A, Beck J (2008) Stochastic subset optimization for optimal reliability problems. Probab Eng Mech 23(2–3):324–338. doi:10.1016/j.probengmech.2007.12.011. http://www.sciencedirect.com/science/article/pii/S0266892007000501. 5th international conference on computational stochastic mechanics

Xu Z, Wu X (2013) A new hybrid stochastic approximation algorithm. Optim Lett 7(3):593–606. doi:10.1007/s11590-012-0443-2

Zeiler MD (2012) ADADELTA: an adaptive learning rate method. arXiv:1212.5701v1 [cs.LG]

Zhu X, Spall JC (2002) A modified second-order SPSA optimization algorithm for finite samples. Int J Adapt Control Signal Process 16:397–409