Recent Advances in SPSA at the Extremes: Adaptive Methods for Smooth Problems and Discrete Methods for Non-Smooth Problems

SGM2014: Stochastic Gradient Methods
IPAM, February 24 – 28, 2014

James C. Spall
The Johns Hopkins University
Applied Physics Laboratory and
Department of Applied Mathematics and Statistics
james.spall@jhuapl.edu

Organization

- We present two key extensions to basic simultaneous perturbation stochastic approximation (SPSA) algorithm
- While SPSA in basic form is not formally a standard stochastic gradient method, it is in same general family of "first-order" SA methods
- We present recent results that deal with problems at the extremes in some sense:
  - **1. SPSA for adaptive estimation in smooth problems, where we wish to obtain a Hessian estimate, and**
  - **2. SPSA-type ideas in fully discrete problems.**

- *Acknowledgment:* The discrete work is joint with Dr. Qi Wang of Johns Hopkins University and Barclays

Extension 1

Adaptive Methods for Smooth Problems

Background

- Interested in finding $\theta = \theta^*$ such that $\boldsymbol{g}(\theta) = \mathbf{0}$ where θ is vector of "adjustables"
- Common special case of minimization: find root $\theta = \theta^*$ to

$$\boldsymbol{g}(\theta) = \frac{\partial L(\theta)}{\partial \theta} = \mathbf{0}$$

  where $L(\theta)$ is scalar-valued loss function
- Assume only (possibly noisy) measurements of $\boldsymbol{g}(\theta)$ and/or $L(\theta)$ available
  - Noisy measurements arise in areas such as Monte Carlo simulation, real-time control/estimation, machine learning, system identification, etc.
- Stochastic approximation (SA) widely used for above root-finding and optimization with noisy measurements, including basic SPSA (*Note:* Google "James Spall NIPS 2012" for video of tutorial)

Brief Diversion (3 slides): Basic SPSA Algorithm (NIPS 2012 tutorial; Spall, 1987, 1992)

- Let $\hat{\boldsymbol{g}}_k(\theta)$ denote SP estimate of $\boldsymbol{g}(\theta)$ at kth iteration
- Let $\hat{\theta}_k$ denote estimate for θ^* at kth iteration
- SPSA algorithm has form

$\hat{g}_k(\hat{\theta}_k)$ is critical !

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

where $\{a_k\}$ is nonnegative gain sequence

- Generic iterative form above is standard in SA; stochastic analogue to steepest descent
- Under conditions, $\hat{\theta}_k \to \theta^*$ in some stochastic sense as $k \to \infty$

Computation of $\hat{g}_k(\bullet)$ (Heart of SPSA)

- Let Δ_k be vector of p independent random variables at kth iteration

$$\Delta_k = \left[\Delta_{k1}, \Delta_{k2}, \ldots, \Delta_{kp}\right]^T$$

- Δ_k typically generated by Monte Carlo
- Let $\{c_k\}$ be sequence of positive scalars
- For iteration $k \rightarrow k+1$, take measurements at design levels: $\hat{\theta}_k \pm c_k \Delta_k$

$$y(\hat{\theta}_k + c_k\Delta_k) = L(\hat{\theta}_k + c_k\Delta_k) + \varepsilon_k^{(+)}$$
$$y(\hat{\theta}_k - c_k\Delta_k) = L(\hat{\theta}_k - c_k\Delta_k) + \varepsilon_k^{(-)}$$

  where $\varepsilon_k^{(\pm)}$ are measurement noise terms

- Common special case is when $\varepsilon_k^{(\pm)} = 0 \; \forall \, k$ (e.g., system identification with perfect measurements of the likelihood function)

Computation of $\hat{\boldsymbol{g}}_k(\bullet)$ (cont'd)

- The standard SP form for $\hat{\boldsymbol{g}}_k(\bullet)$:

$$\hat{g}_k(\hat{\theta}_k) = \begin{bmatrix} \dfrac{y(\hat{\theta}_k + c_k\Delta_k) - y(\hat{\theta}_k - c_k\Delta_k)}{2c_k\Delta_{k1}} \\ \vdots \\ \dfrac{y(\hat{\theta}_k + c_k\Delta_k) - y(\hat{\theta}_k - c_k\Delta_k)}{2c_k\Delta_{kp}} \end{bmatrix}$$

- Note that $\hat{\boldsymbol{g}}_k(\bullet)$ only requires ***two*** measurements of $L(\bullet)$ ***independent*** of ***p***

- Above SP form contrasts with standard finite-difference approximations taking $2p$ (or p+1) measurements

- Intuitive reason why $\hat{\boldsymbol{g}}_k(\bullet)$ is appropriate is that $E[\hat{\boldsymbol{g}}_k(\hat{\theta}_k) \,|\, \hat{\theta}_k] \approx \boldsymbol{g}(\hat{\theta}_k)$; formalized in Spall (1987, 1992)

…Back to Background (cont'd)

- Standard SA algorithms (Robbins-Monro, etc.) show "1st-order" behavior
  - Sharp initial decline (in optimization case)
  - Slow convergence in final phase
  - Sensitivity to units/scaling for elements of θ
- Long-standing interest in capturing "2nd-order" (Newton-like) effects in SA setting
  - Adaptive SA algorithms
  - Iterate averaging
- Shortcomings in both of above in implementation difficulty and practical efficiency

"Standard" Adaptive SPSA Algorithm

- Let $\boldsymbol{G}_k(\theta)$ denote direct (noisy) measurement of $\boldsymbol{g}(\theta)$ or simultaneous perturbation estimate of $\boldsymbol{g}(\theta)$ at kth iteration
- Let $\hat{\theta}_k$ denote estimate for θ^* at kth iteration
- "Standard" adaptive SPSA algorithm has parallel recursions form

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \bar{\boldsymbol{H}}_k^{-1} \boldsymbol{G}_k(\hat{\theta}_k),$$

$$\bar{\boldsymbol{H}}_k = \frac{k}{k+1} \bar{\boldsymbol{H}}_{k-1} + \frac{1}{k+1} \hat{\boldsymbol{H}}_k$$

  where $\{a_k\}$ is nonnegative gain sequence and $\hat{\boldsymbol{H}}_k$ is efficient "per-iteration" Jacobian estimate
- Convergence of $\hat{\theta}_k$ and $\bar{\boldsymbol{H}}_k$ can be shown (Spall, 2000)

Cost of Implementation

- For any *p,* the cost per iteration of adaptive method is

Four *L* measurements
— or —
Three *g* measurements

- Above costs compare very favorably with previous methods:

  $O(p^2)$ loss measurements per iteration in finite-difference setting (e.g., Fabian, 1971)

  $O(p)$ ***g*** measurements per iteration in root-finding setting (e.g., Ruppert, 1985)

Enhanced Adaptive SPSA Algorithm

(Spall, 2009, *IEEE Trans. Auto. Contr.*)

- Standard adaptive algorithm can be improved:
  - Introduce feedback term that helps remove error in per-iteration $\boldsymbol{H}$ estimates
  - Optimal weighting to account for noisy measurements of loss function or of $\boldsymbol{g}(\theta)$
- Enhanced Adaptive SPSA algorithm has same ***general*** parallel recursions form:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \bar{\boldsymbol{H}}_k^{-1} \boldsymbol{G}_k(\hat{\theta}_k),$$

$$\bar{\boldsymbol{H}}_k = (1 - w_k)\bar{\boldsymbol{H}}_{k-1} + w_k\left(\hat{\boldsymbol{H}}_k - \hat{\Psi}_k\right)$$

- Items highlighted in **red** represent new expressions here:

  $\hat{\Psi}_k$ is feedback term

  w_k is optimal weighting

Enhanced Algorithm (cont'd)

- Recursion for θ is unchanged from basic adaptive method
- Critical aspect of recursion for $\boldsymbol{H}$ is the averaging of per-iteration Jacobian estimates $\hat{\boldsymbol{H}}_k$
  - Each per-iteration $\boldsymbol{H}$ estimate formed by simultaneous perturbation of $\boldsymbol{G}_k(\theta)$ values
  - Per-iteration estimate formed from only two $\boldsymbol{G}_k(\theta)$ values (vs. $2\times\dim(\theta)$ values in former adaptive methods)
- Two changes to recursion for $\boldsymbol{H}$ (feedback and weighting)
  - Feedback term uses current (cumulative) estimate of $\boldsymbol{H}$ as true $\boldsymbol{H}$: subtracts out error in $\hat{\boldsymbol{H}}_k$ that depends on true $\boldsymbol{H}$
  - Weighting accounts for increasing effective noise contribution across iterations
  - Special (non-optimal) case of weighting is original adaptive SPSA algorithm where all $\hat{\boldsymbol{H}}_k$ are given equal weight
- Feedback and weighting briefly discussed in slides to follow

Feedback Contribution to Enhanced Algorithm

- Per-iteration Jacobian estimate can be decomposed into four parts:

$$\hat{\boldsymbol{H}}_k = \boldsymbol{H}(\hat{\theta}_k) + \Psi_k + \text{noise} + O(c_k^2)$$

where $\Psi_k = \Psi_k\left(\boldsymbol{H}(\hat{\theta}_k)\right)$ is **error term** and c_k is positive scalar such that $c_k \to 0$ as $k \to \infty$ (c_k is the "difference interval" for the per-iteration estimates)

- Feedback term is best estimate of error term (using current estimate of $\boldsymbol{H}$) at each iteration:

$$\hat{\Psi}_k \equiv \Psi_k\left(\bar{\boldsymbol{H}}_{k-1}\right)$$

- Overall (cumulative) estimate of $\boldsymbol{H}$ (i.e., $\bar{\boldsymbol{H}}_n$, $k = 0,1,\ldots, n$) is based on weighted sums of

$$\hat{\boldsymbol{H}}_k - \hat{\Psi}_k$$

Optimal Weighting

- Recall:

$$\hat{\boldsymbol{H}}_k = \boldsymbol{H}(\hat{\boldsymbol{\theta}}_k) + \boldsymbol{\Psi}_k + \textbf{noise} + O(c_k^2)$$

- The **noise** term is of stochastic order $O(c_k^{-2})$ in the case of noisy loss measurements (optimization) and order $O(c_k^{-1})$ in the case of noisy ***g*** measurements (root-finding)

- Above implies that noise variance ***grows*** with *k*

- Optimal weighting is such that noise growth is "damped down" as *k* gets large

- Can solve for optimal weights via Lagrange multipliers

Comments on Theory

- Former theory on convergence of $\hat{\theta}_k$ continues to apply
- New theory required for convergence of $\bar{\boldsymbol{H}}_k$ due to modified form
- Under various "standard" smoothness and bounded moments conditions, can show $\bar{\boldsymbol{H}}_k \to \boldsymbol{H}(\theta^*)$ a.s.
- For each entry in $\bar{\boldsymbol{H}}_k$, have

$$\frac{\text{variance for enhanced adaptive SPSA}}{\text{variance for standard adaptive SPSA}} < 1$$

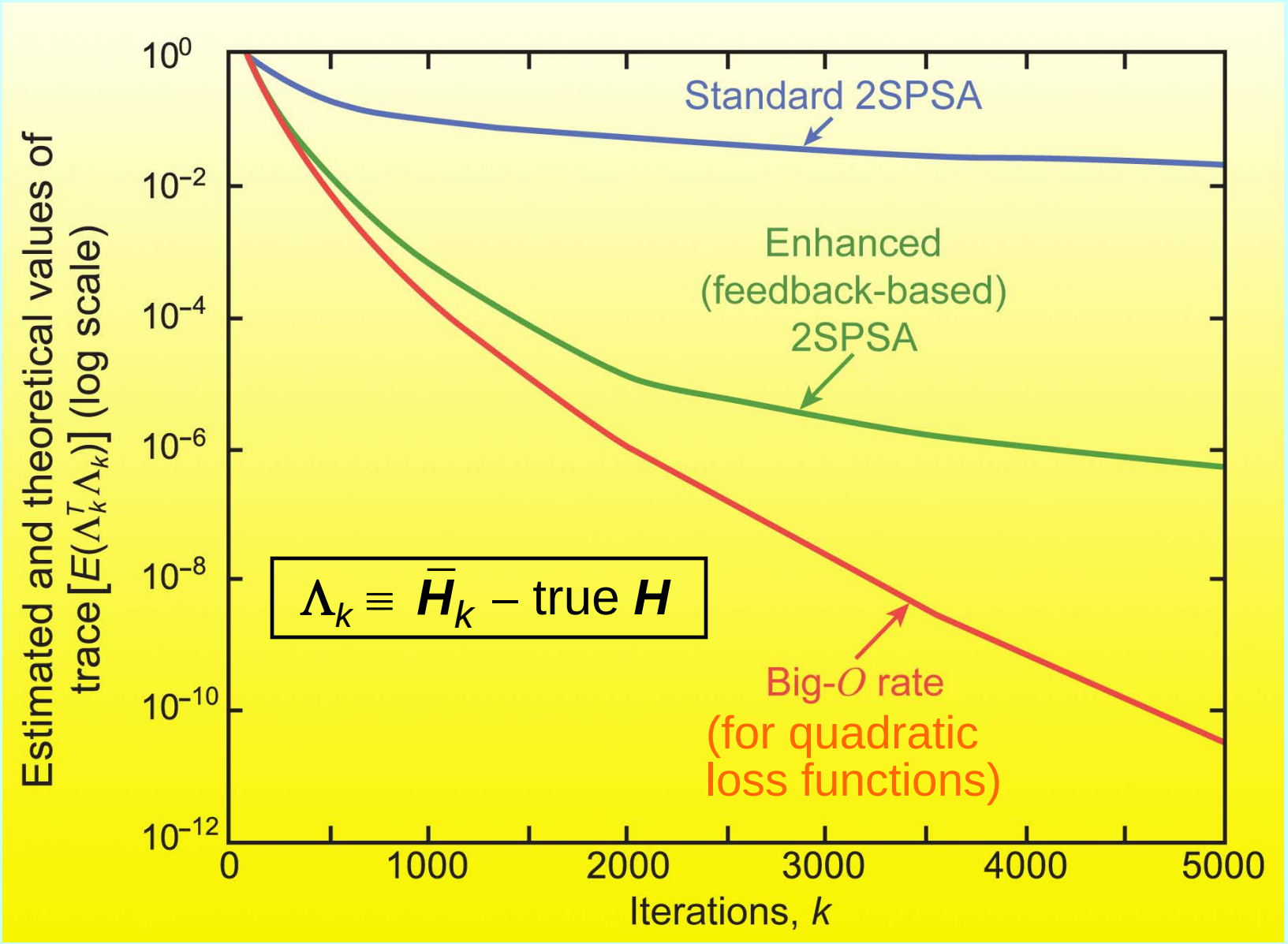
- In special case of noise-free measurements, can show that ***rate*** of convergence of $\bar{\boldsymbol{H}}_k$ to $\boldsymbol{H}(\theta^*)$ is almost $O(e^{-k})$
  - Feedback is critical to fast convergence
  - Applies in case of direct loss or direct $\boldsymbol{g}$ measurements

Numerical Study

- Consider minimization problem based on 4th-order polynomial loss function; dim(θ) = 10
- Compare standard adaptive and enhanced adaptive SPSA in terms of terminal loss function values and ***H*** estimates
- Used the stochastic gradient setting (direct noisy measurements of gradient ***g***) (2SG) and setting with only loss values (2SPSA–see next slide)
- Considered 50 independent runs for each experiment; normalized loss is from 0 (best) to 1(initial condition)

Number of iterations	Normalized loss from standard 2SG	Normalized loss from enhanced 2SG	*P*-values for comparing loss functions
2000	0.019	0.012	0.0061
10,000	0.015	0.0034	0.00049

Relative Convergence Rates for Hessian Estimate in 2SPSA in Noise-Free Problem



Concluding Remarks for Extension 1

- Design of efficient and (relatively) easy to use adaptive search algorithms is long-standing problem
  - Especially difficult in setting of noisy measurements
  - Knowledge of Jacobian/Hessian matrix also useful in contexts outside of improved search (e.g., Cramér-Rao bound)
- Simultaneous perturbation idea can be used to dramatically enhance efficiency in multivariate problems
- Improvements here to "standard" adaptive SPSA are simple to use and further enhance efficiency
  - Feedback to reduce error in "per-iteration" Jacobian/Hessian estimates
  - Optimal weighting to reduce effects of noise
- Using methods for efficient estimation of model of U.S. Navy system

Extension 2

Optimization with Discrete Version of SPSA Using Noisy Loss Function Measurements

(Joint work with Qi Wang of JHU and Barclays)

Discrete Problem Description

- Consider real-valued loss function

$$L(\boldsymbol{\theta}) : \mathbb{Z}^p \rightarrow \mathbb{R}$$

  where $\mathbb{Z}$ is set of integers

- So, $\boldsymbol{\theta}$ assumed to lie on p-dimensional integer grid
- Want to solve problem

$$\min_{\boldsymbol{\theta} \in \mathbb{Z}^p} L(\boldsymbol{\theta})$$

- But we only know the noisy measurements

$$y(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) + \varepsilon(\boldsymbol{\theta})$$

$$E\left(\varepsilon(\boldsymbol{\theta})\right) = 0 \;\; \forall \boldsymbol{\theta}$$

Algorithm Description

- DSPSA: discrete SPSA (Wang and Spall, 2011)
- Step 0: Pick initial guess $\hat{\boldsymbol{\theta}}_0$
- Step 1: Generate $\boldsymbol{\Delta}_k = [\Delta_{k1}, \Delta_{k2}, \ldots, \Delta_{kp}]^T$, where $\boldsymbol{\Delta}_k$ has user-specified distribution satisfying conditions. Special case is when Δ_{ki} are independent Bernoulli random variables ± 1 with probability ½
- Step 2: $\boldsymbol{\pi}(\hat{\boldsymbol{\theta}}_k) = \lfloor \hat{\boldsymbol{\theta}}_k \rfloor + \mathbf{1}_p/2$, where $\lfloor \hat{\boldsymbol{\theta}}_k \rfloor = \left(\lfloor \hat{\theta}_{k1} \rfloor, \ldots, \lfloor \hat{\theta}_{kp} \rfloor \right)$
- Step 3: Evaluate y at $\boldsymbol{\pi}(\hat{\boldsymbol{\theta}}_k) + \boldsymbol{\Delta}_k/2$ and $\boldsymbol{\pi}(\hat{\boldsymbol{\theta}}_k) - \boldsymbol{\Delta}_k/2$
- Step 4: Construct "gradient" approximation:

$$\hat{\boldsymbol{g}}_k(\hat{\boldsymbol{\theta}}_k) = \left[y\left(\boldsymbol{\pi}(\hat{\boldsymbol{\theta}}_k) + \frac{1}{2}\boldsymbol{\Delta}_k \right) - y\left(\boldsymbol{\pi}(\hat{\boldsymbol{\theta}}_k) - \frac{1}{2}\boldsymbol{\Delta}_k \right) \right] \left[\Delta_{k1}^{-1}, \ldots, \Delta_{kp}^{-1} \right]^T$$

- Step 5: After M iterations of recursion, $\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k - a_k \hat{\boldsymbol{g}}_k(\hat{\boldsymbol{\theta}}_k)$ set $\lfloor \hat{\boldsymbol{\theta}}_M \rfloor$ as the solution;

Algorithm Description (Cont'd)

- Remarks:
  - Simple implementation
  - Two loss function measurements in each iteration
  - Make use of the function structure implicitly (gradient-like quantity)
  - Handle noisy measurements
- Convergence property: under some general conditions, the sequence generated by DSPSA converges almost surely to the optimal solution.

Rate of Convergence Results for DSPSA

- We discuss MSE of DSPSA (Wang and Spall, 2013)
- Under some general conditions, we have

$$E\left\|\hat{\boldsymbol{\theta}}_{k+1}-\boldsymbol{\theta}^{*}\right\|^{2} \leq (1-2a_k\mu)E\left\|\hat{\boldsymbol{\theta}}_{k}-\boldsymbol{\theta}^{*}\right\|^{2}+a_k^2 pb \qquad (*)$$

where μ is determined by the loss function and b is a uniform upper bound for

$$E\left[L\left(\boldsymbol{\pi}(\hat{\boldsymbol{\theta}}_k)+\frac{1}{2}\boldsymbol{\Delta}_k\right)-L\left(\boldsymbol{\pi}(\hat{\boldsymbol{\theta}}_k)-\frac{1}{2}\boldsymbol{\Delta}_k\right)\right]^2+E\left(\varepsilon_k^{+}-\varepsilon_k^{-}\right)^2$$

- Note: (*) is a difference-equation-like recursion

Rate of Convergence Results for DSPSA (Cont'd)

- Solving the above difference-equation-like recursion,

$$E\left\|\hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}^*\right\|^2 \leq O\left(e^{-ck^{1-\alpha}}\right) E\left\|\hat{\boldsymbol{\theta}}_0 - \boldsymbol{\theta}^*\right\|^2 + O\left(\frac{1}{k^{\alpha}}\right)$$

  where $c > 0$ and $0.5 < \alpha < 1$

- The first big-*O* term is a function of μ, α, a, A, k and the second big-*O* term is a function of μ, α, a, A, b, k.

Rate of Convergence Results for DSPSA (Cont'd)

- The result on MSE can produce rate of convergence of $P([\hat{\boldsymbol{\theta}}_k] \neq \boldsymbol{\theta}^*)$ to 0 in the big-*O* sense, where $[\hat{\boldsymbol{\theta}}_k]$ indicates the nearest multivariate-integer point of $\hat{\boldsymbol{\theta}}_k$

- Convergence rate of $P([\hat{\boldsymbol{\theta}}_k] \neq \boldsymbol{\theta}^*)$ can be used to compare DSPSA with other standard discrete stochastic algorithms such as stochastic ruler (SR) algorithm (Yan and Mukai, 1992) and stochastic comparison (SC) algorithm (Gong et al, 1999).

Comparison of SR, SC and DSPSA

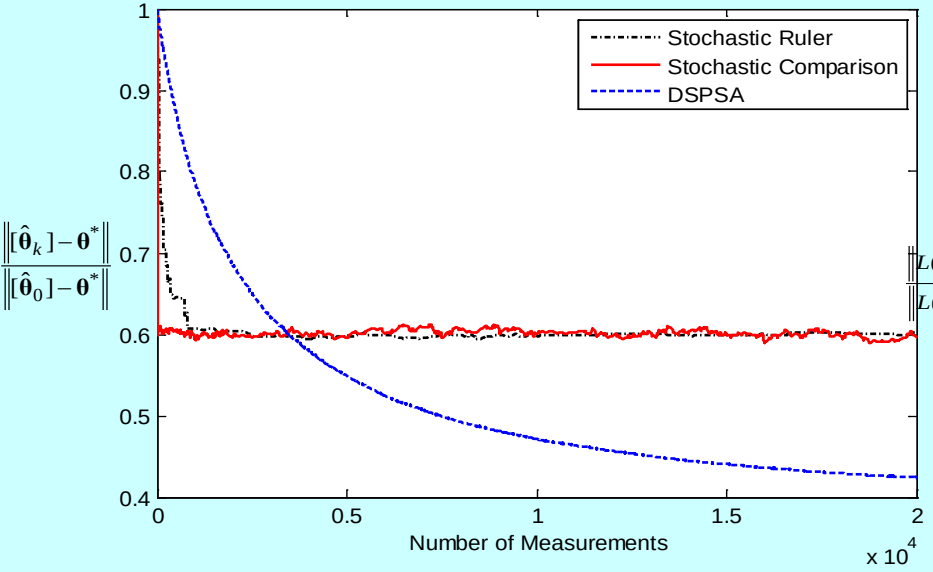
Method Name	Analysis of Rate of Convergence of $P([\hat{\boldsymbol{\theta}}_k] \neq \boldsymbol{\theta}^*)$
DSPSA	$P([\hat{\boldsymbol{\theta}}_k] \neq \boldsymbol{\theta}^*) = O(k^{-\alpha}),\ 0.5 < \alpha \leq 1$
SR	$k^{-\gamma_{SR}} = O(P([\hat{\boldsymbol{\theta}}_k] \neq \boldsymbol{\theta}^*)),\ 0 < \gamma_{SR} \leq 1$
SC	$k^{-\gamma_{SC}} = O(P([\hat{\boldsymbol{\theta}}_k] \neq \boldsymbol{\theta}^*)),,\ 0 < \gamma_{SC} \leq 1$

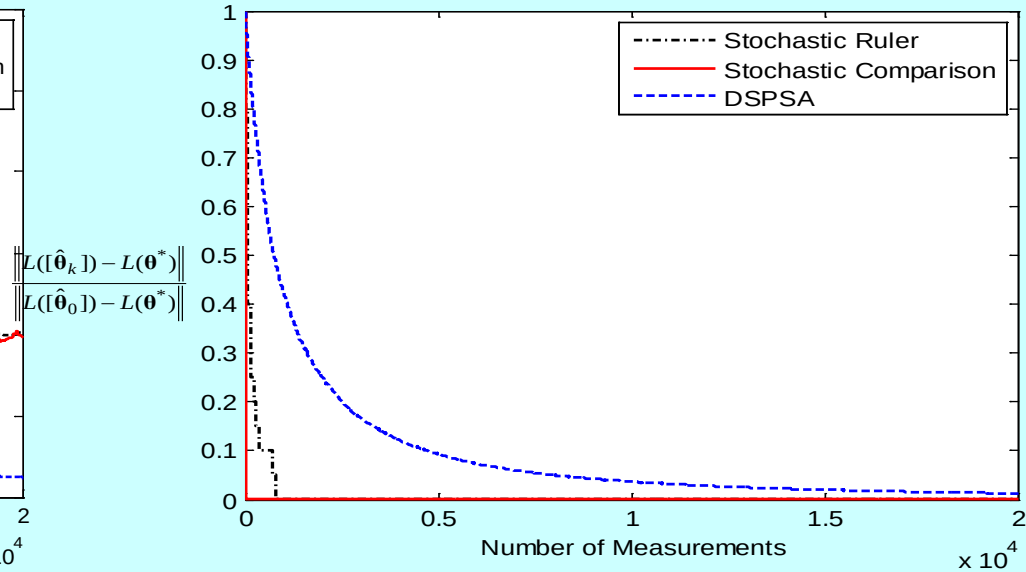
- We find the rate of convergence of SR and SC could not be better than DSPSA.
- The parameters (coefficients) involved in SR and SC are harder to tune than the coefficients in DSPSA.

Numerical Comparisons (one of many, with similar results)

- Skewed quartic loss function (Spall 2003,Ex 6.6) with additive noise $N(0,1)$
- $p\boldsymbol{B}$ is an upper triangular matrix of 1's, $p = 200$, domain $\{-10, -9, \ldots, 9, 10\}^{200}$:

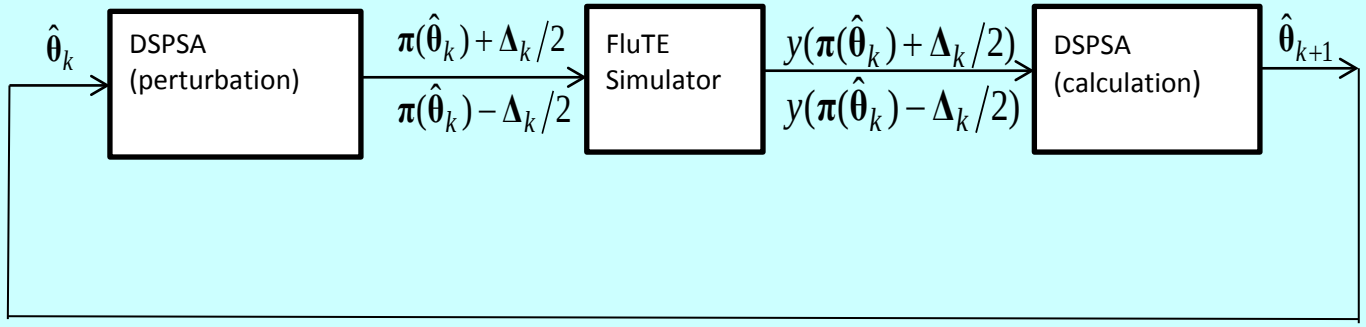
$$L(\boldsymbol{\theta}) = \boldsymbol{\theta}^T \boldsymbol{B}^T \boldsymbol{B}\boldsymbol{\theta} + 0.1\sum_{i=1}^{p}(\boldsymbol{B}\boldsymbol{\theta})_i^3 + 0.01\sum_{i=1}^{p}(\boldsymbol{B}\boldsymbol{\theta})_i^4$$





Application of DSPSA in Resource Allocation in Public Health

- Application of DSPSA towards developing optimal public health strategies for containing the spread of influenza given limited societal resources.
- Open source software for intervention strategies (FluTE: Chao et al., 2010).



Concluding Remarks for Extension 2

- We introduce DSPSA for discrete stochastic optimization problem and show the almost sure convergence property.
- We discuss the rate of convergence of DSPSA, which is $O(1/k^{\alpha})$. Rate of convergence results allow for objective comparison with other stochastic discrete optimization methods (e.g. stochastic ruler and stochastic comparison).
- We consider the application of DSPSA in resource allocation in public health.

Extra Slides

Per-Iteration Jacobian (Hessian) Estimate

$$\hat{\boldsymbol{H}}_k = \frac{1}{2}\left\{\frac{\delta \boldsymbol{G}_k}{2c_k}\left[\Delta_{k1}^{-1}, \Delta_{k2}^{-1}, \ldots, \Delta_{kp}^{-1}\right] + \left(\frac{\delta \boldsymbol{G}_k}{2c_k}\left[\Delta_{k1}^{-1}, \Delta_{k2}^{-1}, \ldots, \Delta_{kp}^{-1}\right]\right)^T\right\}$$

where $\delta \boldsymbol{G}_k = \boldsymbol{G}(\theta + c_k\Delta_k) - \boldsymbol{G}(\theta - c_k\Delta_k)$ and $\boldsymbol{G}(\cdot)$ is direct noisy measurement of $\boldsymbol{g}(\theta)$ or the SP estimate of $\boldsymbol{g}(\theta)$ built from noisy $L(\theta)$ measurements

and

$\Delta_k \equiv [\Delta_{k1}, \Delta_{k2}, \ldots, \Delta_{kp}]^T$ is mean-zero random vector such that the $\{\Delta_{kj}\}$ satisfy standard SPSA conditions (mean zero, symmetrically distributed random variables with finite inverse moments)

Optimal Weighting

- Recall:

$$\bar{\boldsymbol{H}}_k = (1 - w_k)\bar{\boldsymbol{H}}_{k-1} + w_k\left(\hat{\boldsymbol{H}}_k - \hat{\boldsymbol{\Psi}}_k\right)$$

- For the case of noisy loss functions, optimal weights are:

$$w_k = \frac{c_k^4}{\sum_{i=0}^{k} c_i^4}$$

- For the case of noisy root-finding ($\boldsymbol{g}$) measurements, optimal weights are:

$$w_k = \frac{c_k^2}{\sum_{i=0}^{k} c_i^2}$$

- Both of above based on “downweighting” later per-iteration Jacobian estimates to compensate for increased noise contribution