AMO - Advanced Modeling and Optimization, Volume 14, Number 1, 2012

Simulation and the Finite-Difference Stochastic Approximation Method

Miloš Japundžić

Higher School of Professional Business Studies, Novi Sad, Serbia
milos.japundzic@gmail.com

Abstract. An important application of stochastic optimization – simulation optimization, where the objective function can be evaluated only by computer simulation, is considered. We examined some difficulties arised in solving these problems applying the finite-difference stochastic approximation (FDSA) method. Also, we investigated efficiency of the FDSA algorithm depending on the coefficients that generate the step length in optimization algorithm and the perturbation value of gradient approximation of objective function. Efficiency is measured by the mean values of the objective function at the final estimates of the algorithm, over the specified number of replications.

Keywords: Stochastic optimization; Simulation optimization; Efficiency of the FDSA algorithm.

1. INTRODUCTION

There have been countless applications of the stochastic approximation method in the greater than half century since the seminal publication [Robbins and Monro, 1951]. Some areas include neural network, simulation-based optimization, evolutionary algorithms, machine learning, experimental design, and signal processing applications such as noise cancellation and pattern recognition. Building on that paper stochastic approximation (SA) algorithm based on the finite-difference (FD) gradient approximation was introduced, for scalar θ in [Kiefer and Wolfowitz, 1952] and multivariate θ in [Blum, 1954]. In particural, that FDSA algorithm is based on an approximation to the gradient formed from noisy measurements of the objective function. We shall consider efficiency of the algorithm solving the standard unconstrained stochastic optimization problem

$$\min_{\theta} L(\theta) = E\left[y(\theta)\right], \tag{1}$$

where L is a scalar function of n unknowns called objective function, while $y(\theta) = L(\theta) + \varepsilon(\theta)$ is the noisy measurements of objective function ($\varepsilon(\theta)$ represents the noise term). The recursive procedure here is in the general SA form

$$\theta_{k+1} = \theta_k - a_k g_k(\theta_k), \tag{2}$$

[*] **AMO – Advanced Modeling and Optimization. ISSN: 1841-4311.**

where $g_k(\theta_k)$ is the estimation of the gradient $\partial L/\partial\theta$ at the iteration θ_k based on the noisy measurements of objective function, and $a_k > 0$ is the step length in the *k*-th iteration of algorithm. For the estimation of true gradient we use two-sided FD form

$$g_k(\theta_k) = \begin{bmatrix} \dfrac{y(\theta_k + c_k\xi_1) - y(\theta_k - c_k\xi_1)}{2c_k} \\ \vdots \\ \dfrac{y(\theta_k + c_k\xi_n) - y(\theta_k - c_k\xi_n)}{2c_k} \end{bmatrix}, \tag{3}$$

where $\xi_i,\ i = 1,2,..,n,$ denotes a vector with a 1 in the *i*-th place and 0's elsewhere, and $c_k > 0$ defines the difference magnitude.

The paper is organised as follows. In Section 2 we present a set of sufficient conditions for almost sure convergence of the FDSA method, with emphasis on the condition that refers to the step length and the perturbation value of gradient approximation. Section 3 contains analysis of choice of coefficients that generate the step length and the perturbation value of gradient approximation of objective function, where we proposed how to choose these coefficients in order to achieve a better performance of the algorithm. That section also contains numerical results which justify proposed choice of coefficients. All numerical results are obtained using a programming language Matlab. The last section contains final remarks and conclusions.

2. CONVERGENCE OF THE FDSA METHOD

As with any search algorithm, it is of interest to know whether the iteration θ_k generated with FDSA method converges to a solution θ^* as $k \to \infty$. That result guarantees that the iteration θ_k will fall into a small neighborhood of a solution θ^* after sufficient function evaluations. Many sufficient conditions have been given over the years for almost sure convergence of the FDSA recursion in (2) and (3). We shall present so called ″statistics″ conditions.

2.1 Convergence conditions

This subsection presents the set of sufficient conditions for almost sure convergence of the FDSA iterations θ_k [Spall, 2003]. These conditions apply when there is a unique minimum of the problem (1). Hence, they apply if there are no local minima different from the (unique) global minimum. Note that these conditions are sufficient conditions, so many practical implementations of the FDSA will produce satisfactory results when one or more of the conditions are not satisfied. For convenience, let $\varepsilon_k^{(i\pm)} = \varepsilon(\theta_k \pm c_k\xi_i)$.

″Statistics″ conditions:

(C1) **(gain sequences)** $a_k > 0,\ c_k > 0,\ a_k \to 0,\ c_k \to 0,\ \sum_{k=0}^{\infty} a_k = \infty,\ \sum_{k=0}^{\infty} a_k c_k < \infty,$ and $\sum_{k=0}^{\infty} a_k^{\,2}/c_k^{\,2} < \infty.$

(C2) **(unique minimum)** There is a unique minimum θ^* of the problem (1) such that for every $\eta > 0$ $\inf_{\|\theta-\theta^*\|>\eta} \|g(\theta)\| > 0$ and $\inf_{\|\theta-\theta^*\|>\eta} \left[L(\theta) - L(\theta^*)\right] > 0$.

(C3) **(mean-zero and finite variance noise)** For all $i = 1,2,...,n,$ and $k = 0,1,2,...$ $E\left[(\varepsilon_k^{(i+)} - \varepsilon_k^{(i-)})|\Im_k\right] = 0$ a.s. and $E\left[(\varepsilon_k^{(i\pm)2})|\Im_k\right] < C$ a.s. for some $C > 0$ that is independent of k and θ.

(C4) **(bounded Hessian matrix)** The Hessian matrix $H(\theta)$ of objective function exists and is uniformly bounded in norm for all $\theta \in R^n$ (i.e., all componets of $H(\theta)$ are uniformly bounded in magnitude).

From the point of view of the user's input, condition (C1) is the most relevant. This condition includes restrictions on c_k as well as on a_k. It is apparent that $c_k \to 0$ slower than a_k. Also, the condition $\sum_{k=0}^{\infty} a_k c_k < \infty$ requires that a_k and c_k decay faster than sometimes recommended for practical applications. For the sequences $\{a_k\}$ and $\{c_k\}$ the best choice is

$$a_k = \frac{a}{(k+1+A)^{\alpha}} \quad \text{and} \quad c_k = \frac{c}{(k+1)^{\gamma}}, \tag{4}$$

where $a, c, \alpha,$ and γ are strictly positive and the coefficient $A \geq 0$ is stability constant, because it affects the stability of the algorithm.

The problem is how to choose the coefficients a and A in (4), to ensure the convergence of the FDSA. If we choose $A = 0,$ there are a potential problems depending on the size of the coefficient $a.$ Choosing a large numerator a in hope of producing nonnegligible step sizes after the algorithm has been running awhile may cause unstable behavior in the early iterations (when the denominator is still small). On the other hand, choosing a small a leads to stable behavior in the early iterations but sluggish performance in later iterations. For this reason, picking $A > 0$ is usually recommended. A strictly positive A allows choice of a larger a without risking unstable behavior in the early iterations. Then, in the later iterations, the coefficient A in the denominator becomes negligible relative to the $k,$ while the relatively large a in the numerator helps maintain a nonnegligible step size. [Spall, 2003] recommended that a reasonable choice for the stability constant is to pick A such that is approximately 5 to 10 percent of the total number of allowed iterations in the search process.Usually, some numerical experimentations are required to choose the best value of the coefficient a that appears in the gain.

2.2 Rate of convergence

However, convergence by itself gives no information about the rate with which the iterations approach to the solution. For that purpose we need the probability distribution of iterations θ_k, since the iterations generated by FDSA are random vectors. Sacks was the first to establish the asymptotic normality of the FDSA. [Sacks, 1958] shows that for the FDSA algorithm under appropriate conditions

$$\beta \equiv \alpha - 2\gamma > 0 \quad \text{and} \quad 3\gamma - \alpha/2 \geq 0, \tag{5}$$

holds

$$k^{\beta/2}\left(\theta_k-\theta^*\right)\xrightarrow{dist.}N\left(\mu_{FD},\Sigma_{FD}\right),\quad k\to\infty, \tag{6}$$

where $\xrightarrow{dist.}$ denotes converges in distribution, μ_{FD} is a mean vector that depends on the Hessian $H(\theta^*)$ and the third derivative $L'''\left(\theta^*\right)$, Σ_{FD} is some covariance matrix that depends on $H(\theta^*)$, and both μ_{FD} and Σ_{FD} depend on the coefficients $a, c, \alpha,$ and γ in the gain sequences a_k and c_k. The coefficient α and γ govern the decay rate for the gains $\{a_k\}$ and $\{c_k\}$, respectively. (6) implies that for large k, $E\left(\theta_k\right)$ is approximately equal to $\theta^*+k^{-\beta/2}\mu_{FD}$. Hence, $E\left(\theta_k\right)$ has a limiting value of θ^*.

Expresion (6) also implies that the rate at which the iterations θ_k approaches θ^* is proportional in a stochastic sense to $k^{-\beta/2}$ for large k. With the gain forms (4), and under weaker condition (C1')[1] for convergence of the iterations, we know that $\alpha > 1/2$ and $\gamma > 0$. The conditions in (5) put further constraints on α and γ, implying that

$$0.6<\alpha\le 1,\quad 0.1<\gamma<1/2,\quad \alpha-\gamma>1/2. \tag{7}$$

Under the weaker condition (C1') and (4), we find that β is maximized at $\alpha=1$ and $\gamma=1/6$, leading to a maximum rate that is proportional to $k^{-\beta/2}=1/k^{1/3}$, for large k. That is, the maximum rate of convergence for the FDSA algorithm under the general conditions is $O(1/k^{1/3})$ in an appropriate stochastic sense.

3. THE CHOICE OF COEFFICIENTS

This section focuses on the selection of the coefficients $a, A, c, \alpha,$ and γ in the gains a_k and c_k appearing in (4). Suppose that the objective function $L(\theta)$ can only be measured in the presence of the noise $\varepsilon(\theta)$. More specifically, suppose that measurements of $L(\theta)$ at any θ are available as $y_k\left(\theta\right)=L\left(\theta\right)+\varepsilon_k\left(\theta\right)$, $k=0,1,2...$ Estimation θ_k, which is close to the true solution θ^* of the problem (1), in most cases does not have to be the best in the sense of values of $L(\theta)$. This is the reason why efficiency is measured by the mean values of the objective function at the final estimations of solution. The true objective function values $L(\theta_k)$ are used in constructing all tables. These values are not available to the algorithm, which use only noisy measurements $y\left(\theta\right)$ at the various values of θ.

As already mentioned in Section 2, the coefficient A is stability constant, while the coefficients α and γ regulate the decay rate of the gains $\{a_k\}$ and $\{c_k\}$. The rate of convergence of θ_k to θ^* is maximized at $\alpha=1$ and $\gamma=1/6$, but in practical problems it may not be the best to choose those values, because it is often (but not always) preferable in practice to

[1] (C1') is the condition (C1) without $\sum_{k=0}^{\infty}a_kc_k<\infty$.

have a slower decay rate. Practical values of α and γ that are effectively as low as possible while satisfying (C1') and (7) are 0.602 and 0.101, respectively. This provides more power to the algorithm through larger step sizes when k is large.

If the noise has been arised in simulation, than the deviation changes dramatically with θ, so there are a potential problems with negative effects of the noise. In practical applications, the gains are usually chosen by trial and error on some small-scale (reduced number of iterations) version of the full problem. In order to verify reported conclusions a computer program is coded in Matlab to solve the standard test function associate with M/M/1 queueing problem:

Test function :

Let us consider the stochastic optimization problem

$$\min_{\theta\in(0,1)} R(\theta) = \delta/\theta + \eta L(\theta), \tag{8}$$

where $L(\theta)$ denotes the mean number of customers in an M/M/1 queueing system with arrival rate $\lambda = 1$, θ is the mean service time which is to be determined, while δ and η are specified costs. The aim is to find a value of $\theta \in (0,1)$ that minimize total expected cost $R(\theta)$. From queueing theory it is known that $L(\theta) = \theta/(1-\theta)$, so the analytical solution of this problem is

$$\theta^* = \frac{\sqrt{\delta}}{\sqrt{\delta}+\sqrt{\eta}}.$$

In contrast of that, the values of the mean number of customers in system $L(\theta)$ have been obtained by simulation for various values of θ. In this paper $(\delta,\eta) = (10,1)$ is selected, so the corresponding theoretical optimal mean service time is $\theta^* = 0.7597$, and the optimal response is $R(\theta^*) = 16.3246$.

For the initial point of optimization algorithm we choose the mean point of interval (0,1), i.e. we set $\theta_0 = 0.5$. Every time when either iteration θ_k or value $\theta_k \pm c_k$ in the gradient approximation (3) is outside interval $(0,1)$, we choose values which are the nearest to the end points of that interval. In case of our numerical example we shall set those values on 0.01 i 0.95. At each experimental point, the simulation terminates when a number of customers n_{cus} have completed services. In this study, 5 values for n_{cus} have been selected somewhat arbitrarily. For each of these values, the experiment is repeated 10 times to construct confidence intervals. Apparently, as n_{cus} increases the deviation of the estimated mean number of customers in the system decreases (as we can see in Table 1). In other words, n_{cus} serves as an indicator for the deviation of noise.

Table 1: Sample deviation of the estimated mean number of customers in system $L(\theta)$ over the sample of size 3

Number of customers n$_{cus}$	**Sample deviation for various values of θ**					
	$\theta = 0.01$	$\theta = 0.1$	$\theta = 0.2$	$\theta = 0.3$	$\theta = 0.4$	$\theta = 0.5$
100	0.0004	0.0069	0.0364	0.0743	0.0236	0.3853
1000	0.0003	0.0049	0.0091	0.0317	0.0516	0.0209
10000	0.0001	0.0021	0.0058	0.0086	0.0246	0.0288
100000	0.00006	0.0007	0.0002	0.0010	0.0021	0.0235
1000000	0.00001	0.0001	0.0002	0.0009	0.0021	0.0037

Second part of Table 1

Number of customers n$_{cus}$	**Sample deviation for various values of θ**				
	$\theta = 0.6$	$\theta = 0.7$	$\theta = 0.8$	$\theta = 0.9$	$\theta = 0.95$
100	0.9243	0.3099	0.5892	0.3051	2.1028
1000	0.3715	0.5319	1.2178	5.8653	1.8629
10000	0.0647	0.1010	0.0473	0.5874	2.2317
100000	0.0321	0.0319	0.1886	0.5098	0.9930
1000000	0.0060	0.0087	0.0031	0.1183	0.9178

The values $y(\theta)$ appearing in gradient approximation (3) have been obtained by simulation over 3 replications, while coefficients in gain sequences $\{a_k\}$ and $\{c_k\}$ are choosen as folows. For the coefficients α and γ which regulate the decay rate of the gains $\{a_k\}$ and $\{c_k\}$, we choose practical values $\alpha = 0.602$ and $\gamma = 0.101$, while for stability constant A we choose 10% of the total number of allowed iterations in the search process. In order to obtain aproximate optimal values of coefficient a, 5 values of coefficient $\mathrm{c} \in \{0.0001, 0.001, 0.01, 0.1, 1\}$, together with values of mentioned coefficients, have been tested in some small-scale (10 iterations) version of the full problem. Then, those optimal values of coefficients are used in investigating efficiency of the FDSA method depending on the number of function evaluations.

Table 2 presents the mean values of the cost function $R(\theta)$ at the final θ estimates over 10 replications, depending on the number of function evaluations n. The mean values are approximate 90 percent confidence interval constructed according to a ***t***-distribution with 10-1=9 degrees of freedom. These intervals are derived from the sample variance s^2 of the terminal cost function values over the sample of 10 terminal values. Each confidence interval is constructed according to

$$[\text{sample mean} - t_0\sqrt{s^2/10}, \text{ sample mean} + t_0\sqrt{s^2/10}\ ],$$

where t_0 is the *t*-value with 9 degrees of freedom.

Table 2: Sample means and approximate 90 percent confidence intervals for terminal values $R(\theta_k)$ in the case of various number of customers

Number of customers n_{cus}	c	a	**Number of function evaluations n**		
			n=20	n=200	n=1000
100	1	0.0001	16.5647 [16.5642,16.5652]	16.4051 [16.4048,16.4054]	16.3782 [16.3778,16.3785]
1000	0.1	0.01	16.3436 [16.3330,16.3542]	16.3328 [16.3229,16.3428]	16.3281 [16.3258,16.3304]
10000	0.1	0.02	16.3370 [16.3253,16.3488]	16.3277 [16.3261,16.3294]	———————
100000	0.1	0.01	16.3321 [16.3299,16.3342]	16.3272 [16.3269,16.3275]	———————

Analysing the data in Table 2, we can see that for the values $n_{cus} = 100$ and $n_{cus} = 100000$, as number of function evaluations n increases the mean values obtained by algorithm are closer to the optimal response $R(\theta^*) = 16.3246$ (note the nonoverlap in the confidence intervals). The confidence intervals for values $n_{cus} \in \{1000, 10000\}$ illustrate the common phenomenon in stochastic problems that terminal iteration need not to be the best of the iterations, neither in the sense of distance from θ_k to unknown solution θ^*, nor in the sense of values of objective function. Also, in order to neutralize negative effects of the noise arised in simulation (sample deviation significantly varies for various values of θ), the number of customers in simulation should be at least 1000.

4. CONCLUSION

This article provides some suggestions in order to avoid difficulties arised in solving simulation-based optimization problems, when the finite-difference stochastic approximation (FDSA) method is used for solving. Because the deviation of the noise arised in simulation significantly varies from iteration to iteration, there are problems in choosing the values of coefficients that generate the step length in optimization algorithm and the perturbation value of gradient approximation of objective function. This is the reason why it is difficult to automate the gain selection process, so the values of coefficients (especially values of coefficients c and a) must be chosen by trial and error.

At the end, note that simulation process is just the part of the optimization process, so as the number of customers in simulation process increases computer time also significantly increases. This is the reason why in Table 2 the mean values for $n = 1000$ function evaluations, in cases $n_{cus} = 10000$ and $n_{cus} = 100000$, are omitted.

5. APPENDIX

```
function FDSA_queueing
%This code computes the mean values of objective function at the terminal iterations for the FDSA method, as well as corresponding confidence intervals

global p      %global variables
p=1;          %dimension of problem
n=1000;          %number of function evaluations
replications=10;     %number of replications
theta_0=0.5*ones(p,1);    %initial point

a=1;                 %values of coefficients
c=1;
A=0.1*n/(2*p);
alfa=0.602;
gama=0.101;

g=zeros(p,1);
true_theta=0.7597;  %true solution
theta_lo=0.01*ones(p,1);     %lower bounds on theta
theta_hi=0.95*ones(p,1);     %upper bounds on theta
rand('seed',71111113)
fun_noise='queueing'; %values in the presence of noise
fun_nonoise='objective_function'; %true objective function
meanfun=0;
meanfunsq=0;
t=1.833;               %t-value for Student’s distribution

    for i=1:replications
          theta=theta_0;
          for k=1:n/(2*p)
                ak=a/(k+A)^alfa;
                ck=c/k^gama;
                thetaplus=theta+ck;
                thetaminus=theta-ck;
                thetaplus=min(thetaplus,theta_hi);
                thetaplus=max(thetaplus,theta_lo);
                thetaminus=min(thetaminus,theta_hi);
                thetaminus=max(thetaminus,theta_lo);
                yplus=feval(fun_noise,thetaplus);
                yminus=feval(fun_noise,thetaminus);
                g=(yplus-yminus)/(2*ck);
              theta=theta-ak*g;
              theta=min(theta,theta_hi);
              theta=max(theta,theta_lo);
        end
        eval=feval(fun_nonoise,theta);
        meanfun=(i-1)*meanfun/i+eval/i;
        meanfunsq=(i-1)*meanfunsq/i+(eval^2)/i;
    end

disp('Sample mean')
meanfun

if replications >1
           s=(replications/(replications-1))^0.5*(meanfunsq-
              meanfun^2)^0.5;
           int_left= meanfun - t*s/replications^0.5;
           int_right= meanfun + t*s/replications^0.5;
           disp('Confidence interval')
           [int_left,int_right]
else
end
```

REFERENCES

Blum, J.R., (1954) Multidimensional Stochastic Approximation Methods, Annals of Mathematical Statistics, vol. 25, pp. 737-744.

Fu, M.C., (2002) Optimization for Simulation: Theory vs. Practice, INFORMS Journal of Computing, vol.14, pp. 192-215.

Fu, M.C., (1994) Optimization via Simulation: A Review, Annals of Operational Research, vol. 53, pp. 199-248.

Japundžić, M., (2010) Efficiency of the Modifications of Deterministic Methods in Solving the Stochastic Optimization Problems, MSc Thesis (In Serbian language), University of Novi Sad, Faculty of Sciences and Mathematics.

Japundžić, M., (2012) Efficiency of the Stochastic Approximation Method, Yugoslav Journal of Operations Research, vol. 22, No 1: Future Issue, OnLine-First.

Kao, C., Song, W.T., Chen, S.P., (1997) A Modified Quasi-Newton Method for Optimization in Simulation, International Transactions in Operational Research,vol.4, pp. 223-233.

Kao, C., Chen, S.P., (2006) A Stochastic Quasi-Newton Method For Simulation Response Optimization, European Journal of Operational Research, vol.173, pp. 30-46.

Kiefer, J., Wolfowitz, J., (1952) Stochastic Estimation of a Regression Function, Annals of Mathematical Statistics, vol.23, pp. 462-466.

Law, A.M., Kelton, W.D., (2000) Simulation Modeling and Analysis, McGraw-Hill, New York.

L'Ecuyer, P., Giroux, N., Glynn, P.W., (1994) Stochastic Optimization by Simulation: Numerical Experiments with the M/M/1 Queue in Steady-state, Management Science, vol.40, pp. 1245-1261.

Nocedal, J., Wright, S., (1999) Numerical Optimization, Springer, New York.

Robbins, H., Monro, S., (1951) A Stochastic Approximation Method, Annals of Mathematical Statistics, vol.22, pp. 400-407.

Sheldon, R.M., (2002) Introduction to Probability Models, Academic Press.

Sacks, J. (1958) Asymptotic Distribution of Stochastic Approximation Procedures, Annals of Mathematical Statistics, vol.29, pp. 373-405.

Kleywegt, J.A., Shapiro, A., (2000) Stochastic Optimization, Chapter 101, School of Industrial and Systems Engineering.

Spall, J.C. (2003) Introduction to Stochastic Search and Optimization, Wiley-Interscience, New Jersey.

Winston, L.W. , (1994) Operations Research: Applications and Algorithms, Duxbury Press.