Stochastic Optimization for Adaptive Labor Staffing in Service Systems

L.A. Prashanth[1], H.L. Prasad[1], Nirmit Desai[2],
Shalabh Bhatnagar[1], and Gargi Dasgupta[2]

[1] Indian Institute of Science, Bangalore, India
{prashanth,hlprasu,shalabh}@csa.iisc.ernet.in
[2] IBM Research, Bangalore, India
{nirmdesa,gaargidasgupta}@in.ibm.com

Abstract. Service systems are labor intensive. Further, the workload tends to vary greatly with time. Adapting the staffing levels to the workloads in such systems is nontrivial due to a large number of parameters and operational variations, but crucial for business objectives such as minimal labor inventory. One of the central challenges is to optimize the staffing while maintaining system steady-state and compliance to aggregate SLA constraints. We formulate this problem as a parametrized constrained Markov process and propose a novel stochastic optimization algorithm for solving it. Our algorithm is a multi-timescale stochastic approximation scheme that incorporates a SPSA based algorithm for 'primal descent' and couples it with a 'dual ascent' scheme for the Lagrange multipliers. We validate this optimization scheme on five real-life service systems and compare it with a state-of-the-art optimization tool-kit OptQuest. Being two orders of magnitude faster than OptQuest, our scheme is particularly suitable for adaptive labor staffing. Also, we observe that it guarantees convergence and finds better solutions than OptQuest in many cases.

Keywords: Service systems, labor optimization, constrained stochastic optimization.

1 Introduction

In service-based economies, the clients and service providers exchange value through service interactions and reach service outcomes. Service requests of a client can vary greatly in the skills required to fulfill the request, expected turn-around time, and the context of the client's business needs. As a result, service delivery is a labor-intensive business and it is crucial to optimize labor costs. A *Service System (SS)* is an organization composed of (i) the resources that support, and (ii) the processes that drive service interactions so that the outcomes meet customer expectations [1]. The contributions of this paper are focused on data-center management services but can be extended to all service domains. The service providers manage the data-centers from remote locations called *delivery centers* where groups of *service workers (SW)* skilled in specific technology areas support corresponding *service requests (SR)*. In each group, the processes, the people and the customers that drive the operation of center constitute a SS. A delivery center is a system of multiple SS. A central component in these operational models

G. Kappel, Z. Maamar, H.R. Motahari-Nezhad (Eds.): ICSOC 2011, LNCS 7084, pp. 487–494, 2011.

is the policy for assigning SRs to SWs, called the *dispatching policy*. The fundamental challenges here are: (i) given an SS with its operational characteristics, the staffing across skill levels and shifts needs to be optimized while maintaining steady-state and compliance to Service Level Agreement (SLA) constraints. (ii) the SS characteristics such as work patterns, technologies and customers supported change frequently, and hence the optimization needs to be adaptive.

This paper presents a novel stochastic optimization algorithm *SASOC (Staff Allocation using Stochastic Optimization with Constraints)* to address the above challenges. SASOC is a three timescale stochastic approximation scheme that uses SPSA-based [2] estimates for performing gradient descent in the primal, while having a dual ascent scheme for the Lagrange multipliers. In the evaluation step of SASOC, we leverage the simulation-based operational models developed in [3] for two of the dispatching policies, namely, PRIO-PULL (basic priority scheme) and EDF (earliest deadline first). We evaluate our algorithm on data from five real-life SS in the data-center management domain. In comparison with the state-of-the-art OptQuest optimization toolkit [4], we find that (a) SASOC is two orders of magnitude faster than OptQuest, (b) it finds solutions of comparable quality to OptQuest, and (c) it guarantees convergence where OptQuest does not find feasibility even with 5000 iterations. Precisely due to the guaranteed convergence and by returning good solutions quickly, SASOC is well suited to better address the above two challenges, especially with respect to adaptivity. By comparing SASOC results on two independent operational models corresponding to PRIO-PULL and EDF dispatching policies, we show that SASOC's performance is independent of the operational model of SS.

We now review relevant literature in service systems and stochastic optimization. In [5], a two step mixed-integer program is formulated for the problem of dispatching SRs within service systems. While their goal is similar, their formulation does not model the stochastic variations of arrivals or processing times. Further, unlike our framework, the SLA constraints in their formulation cannot be aggregates. In [6], the authors propose a scheme for shift-scheduling in the context of third-level IT support systems. Unlike this paper, they do not validate their method against data from real-life third-level IT support. In [3], a simulation framework for evaluating dispatching policies is proposed. A scatter search technique is used to search over the space of SS configurations and optimize the staff there. While we share their simulation model, the goal in this paper is to propose a fundamentally new algorithm that is based on stochastic optimization. In general, none of the above papers propose an optimization algorithm that is geared for SS and that leverages simulation to adapt optimization search parameters.

A popular and highly efficient simulation based local optimization scheme for gradient estimation is Simultaneous Perturbation Stochastic Approximation (SPSA) proposed by [7]. SPSA is based on the idea of randomly perturbing the parameter vector using i.i.d., symmetric, zero-mean random variables that has the critical advantage that it needs only two samples of the objective function for any N-dimensional parameter. Usage of deterministic perturbations instead of randomized was proposed in [2]. The deterministic perturbations there were based either on lexicographic or Hadamard matrix based sequences and were found to perform better than their randomized perturbation counterparts. In [8], several simulation based algorithms for constrained optimization

have been proposed. Two of the algorithms proposed there use SPSA for estimating the gradient, after applying Lagrangian relaxation procedure to the constrained optimization problem.

2 Problem Formulation

We consider the problem of finding the optimal number of workers for each shift and of each skill level in a SS, while adhering to a set of SLA constraints. We formulate this as a constrained optimization problem with the objective of minimizing the labor cost in the long run average sense, with the constraints being on SLA attainments. The underlying dispatching policy (that maps the service requests to the workers) is fixed and is in fact, parametrized by the set of workers. In essence, the problem is to find the 'best' parameter (set of workers) for a given dispatching policy.

In a typical SS, the arrival as well as service time of SRs are probabilistic. Thus, the system can be modeled to be evolving probabilistically over states, where each state transition incurs a cost. The objective is to minimize the long run average sum of this single stage cost, while adhering to a set of SLA constraints. The state is the vector of the utilization of workers for each shift and skill level, and the current SLA attainments for each customer and each SR priority. Any arriving SR has a customer identifier and a priority identifier. Let A be the set of shifts of the workers, B be the set of skill levels of the workers, C be the set of all customers and P be the set of all possible priorities in the SS under consideration. The state X_n at time n is given by

$$X_n = (u_{1,1}(n), ..., u_{|A|,|B|}(n), \gamma'_{1,1}(n), ..., \gamma'_{|C|,|P|}(n), q(n)),$$

where $0 \leq u_{i,j} \leq 1$ is the per-unit utilization of the workers in shift i and skill level j. $0 \leq \gamma'_{i,j} \leq 1$ denotes the SLA attainment level for customer i and priority j. q is a Boolean variable that denotes the queue feasibility status of the system at instant n. In other words, q is false if the growth rate of the SR queues (for each complexity) is beyond a threshold and true otherwise. We need q to ensure system steady-state which is independent of SLA attainments because SLA attainments are computed only on the SRs that were completed and not on SRs queued up in the system. The action a_n at instant n specifies the number of workers of each skill level in each shift.

The single stage cost function is designed so as to minimize the under-utilization of workers as well as over-achievement/under-achievement of SLAs. Here, under-utilization of workers is the complement of utilization and in essence, this is equivalent to maximizing the worker utilizations. The over-achievement/under-achievement of SLAs is the distance between attained SLAs and the contractual SLAs. Hence, the cost function is designed to balance between two conflicting objectives and has the form:

$$c(X_n) = r \times \left(1 - \sum_{i=1}^{|A|} \sum_{j=1}^{|B|} \alpha_{i,j} \times u_{i,j}(n)\right) + s \times \left(\sum_{i=1}^{|C|} \sum_{j=1}^{|P|} \left|\gamma'_{i,j}(n) - \gamma_{i,j}\right|\right), \quad (1)$$

where $r, s \geq 0$ and $r + s = 1$. $0 \leq \gamma_{i,j} \leq 1$ denotes the contractual SLA for customer i and priority j. Note that the first factor uses a weighted sum of utilizations over workers

from each shift and across each skill level. The weights $\alpha_{i,j}$ are derived from the workload distribution across shifts and skill levels over a month long period. These weights satisfy $0 \leq \alpha_{i,j} \leq 1, \quad \sum_{i=1}^{|A|} \sum_{j=1}^{|B|} \alpha_{i,j} = 1$. This prioritization of workers helps in optimizing the worker set based on the workload.

The constraints are on the SLA attainments and are given by:

$$g_{i,j}(X_n) = \gamma_{i,j} - \gamma'_{i,j}(n) \leq 0, \forall i = 1, \ldots, |C|, j = 1, \ldots, |P|, \tag{2}$$

$$h(X_n) = 1 - q(n) \leq 0, \tag{3}$$

where constraints (2) specify that the attained SLA levels should be equal to or above the contractual SLA levels for each pair of customer and priority. The constraint (3) ensures that the SR queues for each complexity in the system stay bounded.

Considering that the state is a vector of utilization levels and the current SLA attainments, it is easy to see that $\{X_n, n \geq 1\}$ is a constrained Markov process for any given dispatch policy. Further, $\{X_n, n \geq 1\}$ is parametrized with

$$\theta = (W_{1,1}, ..., W_{|A|,|B|})^T \in \mathcal{R}^{|A| \times |B|},$$

where $W_{i,j}$ indicates the number of service workers whose skill level is j and whose shift index is i. We want to find an optimal value for the parameter vector θ that minimizes the long-run average sum of single-stage cost $c(X_n)$ while maintaining queue stability $h(X_n)$ and compliance to contractual SLAs $g_{i,j}(X_n)$, $\forall i = 1, \ldots, |C|, j = 1, \ldots, |P|$.

We let the parameter vector $\theta \in \mathcal{R}^{|A| \times |B|}$ take values in a compact set $M \stackrel{\triangle}{=} [0, W_{\max}]^{|A| \times |B|}$ through the projection operator Π defined by $\Pi(\theta) \stackrel{\triangle}{=} (\pi(W_{1,1}), \ldots, \pi(W_{|A|,|B|}))^T$, $\theta \in \mathcal{R}^{|A| \times |B|}$. Here $\pi(x) \stackrel{\triangle}{=} \min(\max(0, x), W_{\max})$. In essence, the projection operator Π keeps each $W_{i,j}$ bounded between 0 and $W_{\max}$ and this is necessary for ensuring the convergence of θ. Our aim is to find a θ that minimizes the long run average cost,

$$\begin{aligned} &J(\theta) \stackrel{\triangle}{=} \lim_{n \to \infty} \tfrac{1}{n} \sum_{m=0}^{n-1} E[c(X_m)] \\ &\text{subject to} \\ &G_{i,j}(\theta) \stackrel{\triangle}{=} \lim_{n \to \infty} \tfrac{1}{n} \sum_{m=0}^{n-1} E[g_{i,j}(X_m)] \leq 0 \quad \forall i = 1, \ldots, |C|, j = 1, \ldots, |P|, \\ &H(\theta) \stackrel{\triangle}{=} \lim_{n \to \infty} \tfrac{1}{n} \sum_{m=0}^{n-1} E[h(X_m)] \leq 0 \end{aligned} \tag{4}$$

Here each step from n to $n+1$ indicates a state transition from X_n to X_{n+1}, incurring a cost $c(X_n)$. The parameter θ decides what cost is incurred and whether the constraints are met. The actions a_j are assumed to be governed by the underlying dispatching policy of the SS. We make a standard assumption that the Markov process $\{X_n, n \geq 1\}$ is ergodic for the given dispatching policy, which is true in general. Thus, the limits in the above optimization problem are well defined. While it is desirable to find the optimum $\theta^* \in M$ i.e.,

$$\theta^* = \text{argmin} \left\{ J(\theta) \text{ s.t. } \theta \in M, G_{i,j}(\theta) \leq 0, i = 1, \ldots, |C|, j = 1, \ldots, |P|, H(\theta) \leq 0 \right\},$$

it is in general very difficult to achieve a global minimum. We apply the Lagrangian relaxation procedure to the above problem and then use a local optimization scheme based on SPSA for finding the optimum parameter θ^*.

3 Our Algorithm (SASOC)

The constrained optimization problem (4) can be expressed using the standard Lagrange multiplier theory as an unconstrained optimization problem given below.

$$\max_{\lambda} \min_{\theta} L(\theta, \lambda) \stackrel{\triangle}{=} \lim_{n \to \infty} \frac{1}{n} \sum_{m=0}^{n-1} E \left\{ c(X_m) + \sum_{i=1}^{|C|} \sum_{j=1}^{|P|} \lambda_{i,j} g_{i,j}(X_m) + \lambda_f h(X_m) \right\} \tag{5}$$

where $\lambda_{i,j} \geq 0, \quad \forall i = 1, \ldots, |C|, j = 1, \ldots, |P|$ represent the Lagrange multipliers corresponding to constraints $g_{i,j}(\cdot)$ and λ_f represents the Lagrange multiplier for the constraint $h(\cdot)$, in the optimization problem (4). The function $L(\theta, \lambda)$ is commonly referred to as the Lagrangian. An optimal (θ^*, λ^*) is a saddle point in the Lagrangian i.e. $L(\theta, \lambda^*) \geq L(\theta^*, \lambda^*) \geq L(\theta^*, \lambda)$. Thus, it is necessary to design an algorithm which descends in θ and ascends in λ to find the optimum point. The simplest iterative procedure for this purpose would use the gradient of the Lagrangian with respect to θ and λ to descend and ascend respectively. However, for the given system the computation of gradient with respect to θ would be intractable due to lack of a closed form expression of the Lagrangian. Thus, a simulation based algorithm is required. We employ an SPSA technique [7,2] for obtaining a stochastic approximation descent procedure for θ. For $\lambda_{i,j}$ and λ_f, values of $g_{i,j}(\cdot)$ and $h(\cdot)$ respectively can be seen to provide a stochastic ascent direction.

The above explanation suggests that an algorithm would need three stages in each of its iterations. (i) The inner-most stage which performs one or more simulations over several time steps; (ii) The next outer stage which computes a gradient estimate using simulation results of the inner most stage and then updates θ along descent direction. This stage would perform several iterations for a given λ and find out the best θ; and (iii) The outer-most stage which computes the long-run average value of each constraint using the iterations in the inner two stages and updates the Lagrange multipliers λ for using that as the ascent direction. The above three steps need to be performed iteratively till the solution converges to a saddle point described previously. However, this approach suffers from a serious drawback of requiring to perform several simulations as a whole as one outer stage update happens for one full run of inner stages at both levels. This issue gets addressed by using simultaneous updates to all three stages but with different time-steps, the outer-most having the smallest while the inner-most having the largest time-steps. This comes under the realm of multiple time-scale stochastic approximation [9, Chapter 6]. We develop a three time-scale stochastic approximation algorithm that does primal descent using an SPSA based actor-critic algorithm while performing dual ascent on the Lagrange multipliers. The update rule for SASOC is given below:

$$
\left.
\begin{aligned}
&W_{i,j}(n+1) = \Pi\left(W_{i,j}(n) + b(n)\left(\frac{\bar{L}(nK)-\bar{L}'(nK)}{\delta\triangle_{i,j}(n)}\right)\right), \\
&\qquad\qquad \forall i = 1,2,\ldots,|A|, j = 1,2,\ldots,|B|, \\
&\text{where for } m = 0,1,\ldots,K-1, \\
&\bar{L}(nK+m+1) = \bar{L}(nK+m)+ \\
&d(n)(c(X_{nK+m}) + \sum_{i=1}^{|C|}\sum_{j=1}^{|P|}\lambda_{i,j}(nK)g_{i,j}(X_{nK+m}) + \lambda_f h(X_{nK+m}) - \bar{L}(nK+m)), \\
&\bar{L}'(nK+m+1) = \bar{L}'(nK+m)+ \\
&d(n)(c(\hat{X}_{nK+m}) + \sum_{i=1}^{|C|}\sum_{j=1}^{|P|}\lambda_{i,j}(nK)g_{i,j}(\hat{X}_{nK+m}) + \lambda_f h(\hat{X}_{nK+m}) - \bar{L}'(nK+m)), \\
&\lambda_{i,j}(n+1) = (\lambda_{i,j}(n) + a(n)g_{i,j}(X_n))^+, \forall i = 1,2,\ldots,|C|, j = 1,2,\ldots,|P|, \\
&\lambda_f(n+1) = (\lambda_f(n) + a(n)h(X_n))^+,
\end{aligned}
\right\} \quad (6)
$$

where

- $\hat{X}_m$ represents the state at iteration m from the simulation run with perturbed parameter $\theta_{[\frac{n}{K}]} + \delta\Delta_{[\frac{n}{K}]}$. Here $[\frac{n}{K}]$ denotes the integer portion of $\frac{n}{K}$. For simplicity, hereafter we use $\theta + \delta\Delta$ to denote $\theta_{[\frac{n}{K}]} + \delta\Delta_{[\frac{n}{K}]}$;
- $\delta > 0$ is a fixed perturbation control parameter while Δ represents a deterministic perturbation sequence chosen according to an associated Hadamard matrix, explained later in this section;
- The operator $\Pi(\cdot)$ ensures that the updated value for θ stays within the chosen compact space T;
- $\bar{L}$ and $\bar{L}'$ represent Lagrangian estimates for θ and $\theta + \delta\Delta$ respectively. Thus, for each iteration two simulations are carried out, one with θ parameter and the other with the perturbed parameter $\theta + \delta\Delta$, the result of which is used to update $\bar{L}$ and $\bar{L}'$; and
- $K \geq 1$ is a fixed parameter which controls the rate of update of θ in relation to that of $\bar{L}$ and $\bar{L}'$. This parameter allows for accumulation of updates to $\bar{L}$ and $\bar{L}'$ for K iterations in between two successive θ updates.

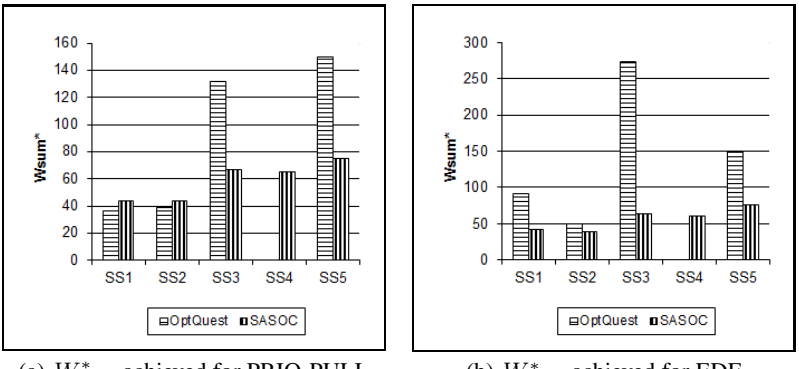
The step-sizes $\{a(n)\}$, $\{b(n)\}$ and $\{d(n)\}$ satisfy $\sum_n a(n) = \sum_n b(n) = \sum_n d(n) = \infty; \sum_n (a^2(n) + b^2(n) + d^2(n)) < \infty, \frac{b(n)}{d(n)}, \frac{a(n)}{b(n)} \rightarrow 0$ as $n \rightarrow \infty$.. The above choice of step-size ensure separation of time-scales between the recursions of $W_{i,j}, \bar{L}, \bar{L}'$ and λ. The perturbation sequence $\{\triangle(n)\}$ is constructed using Hadamard matrices and the reader is referred to Lemma 3.3 of [2] for details of the construction.

4 Simulation Experiments

We use the simulation framework developed in [3] and focus on the PRIO-PULL and EDF dispatching policies. However, SASOC algorithm is agnostic to the dispatching policies. In PRIO-PULL policy, SRs are queued in the complexity queues based directly

on the priority assigned to them by the customers. On the other hand, in the EDF policy the time left to SLA target deadline is used to assign the SRs to the SWs i.e., the SW works on the SR that has the earliest deadline. We implemented our SASOC algorithm as well as an algorithm for staff allocation using the state-of-the-art optimization toolkit OptQuest. OptQuest is a well-established tool for solving simulation optimization problems [4].

For the SASOC algorithm, we set the weights in the single-stage cost function $c(X_m)$, see (1), as $r = s = 0.5$. We thus give equal weightage to both the worker utilization and the SLA over-achievement components. The feasibility Boolean variable q used in the constraint (3) was set to false (i.e., infeasible) if the queues were found to grow by 1000% over a two-week period. On each SS, we compare our SASOC algorithm with the OptQuest algorithm using W_{sum} as the performance metric. Here $W_{sum} \triangleq \sum_{i=1}^{|A|} \sum_{j=1}^{|B|} W_{i,j}$ is the sum of workers across shifts and skill levels. We observe that simulation run-times are proportional to the number of SS simulations and hence, an order of magnitude higher for OptQuest as compared to SASOC.



(a) W^*_{sum} achieved for PRIO-PULL (b) W^*_{sum} achieved for EDF

Fig. 1. Performance of OptQuest and SASOC for two different dispatching policies on five real SS (*Note:* OptQuest is infeasible over SS4)

Figs 1(a) and 1(b) compare the W^*_{sum} achieved for OptQuest and SASOC algorithms using PRIO-PULL on five real life SS. Here W^*_{sum} denotes the value obtained upon convergence of W_{sum}. On three SS pools, namely SS3, SS4 and SS5, respectively, we observe that our SASOC algorithm finds a significantly better value of W^*_{sum} as compared to OptQuest. Note in particular that the performance difference between SASOC, and OptQuest on SS3 and SS5 is nearly 100%. Further, on SS4, OptQuest is seen to be infeasible whereas SASOC obtains a feasible good allocation. On the other two pools, SS1 and SS2, OptQuest is seen to be slightly better than SASOC. Further, the SASOC algorithm requires 500 iterations, with each iteration having 20 replications of the SS - 10 each with unperturbed parameter θ and perturbed parameter $\theta + \delta\Delta$ respectively, whereas OptQuest requires 5000 iterations with each iteration of 100 replications. This implies a two orders of magnitude improvement while searching for the optimal SS configuration in SASOC as compared to OptQuest. Fig 1(b) presents similar results for

the case of EDF dispatching policy. The behavior of OptQuest and SASOC algorithms was found to be similar to that of PRIO-PULL and SASOC shows significant performance improvements over OptQuest here as well. We observe that SASOC is a robust algorithm that gives a reliably good performance, is computationally efficient and is provably convergent, unlike OptQuest that does not possess these features.

5 Conclusions

We presented an efficient algorithm SASOC for optimizing staff allocation in the context of SS. We formulated the problem as a constrained optimization problem where both the objective and constraint functions were long run averages of a state dependent single-stage cost function. A novel single stage cost that balanced the conflicting objectives of maximizing worker utilizations and minimizing the over-achievement of SLA was employed. Numerical experiments were performed to evaluate SASOC against prior work in the context of a real-life service system. SASOC showed much superior performance compared to the state-of-the-art simulation optimization toolkit OptQuest, as it (a) was an order of magnitude faster than OptQuest, (b) found solutions of quality comparable to those found by OptQuest even in scenarios where OptQuest did not find feasibility even after 5000 iterations. By comparing SASOC results on two independent operational models, we showed that SASOCs performance is independent of the operational model of SS. We are in the process of developing and applying a second-order Newton based scheme with SPSA estimates for this problem. It would be of interest to compare the performance of that scheme with the one proposed here. It would also be of interest to prove the theoretical convergence of these algorithms.

References

1. Alter, S.: Service system fundamentals: Work system, value chain, and life cycle. IBM Systems Journal 47(1), 71–85 (2008)
2. Bhatnagar, S., Fu, M., Marcus, S., Wang, I.: Two-timescale simultaneous perturbation stochastic approximation using deterministic perturbation sequences. ACM Transactions on Modeling and Computer Simulation (TOMACS) 13(2), 180–209 (2003)
3. Banerjee, D., Desai, N., Dasgupta, G.: Simulation-based evaluation of dispatching policies in service systems. In: Winter Simulation Conference (2011) (under review)
4. Laguna, M.: Optimization of complex systems with optquest. OptQuest for Crystal Ball User Manual, Decisioneering (1998)
5. Verma, A., Desai, N., Bhamidipaty, A., Jain, A., Nallacherry, J., Roy, S., Barnes, S.: Automated optimal dispatching of service requests. In: SRII Global Conference (2011)
6. Wasserkrug, S., Taub, S., Zeltyn, S., Gilat, D., Lipets, V., Feldman, Z., Mandelbaum, A.: Creating operational shift schedules for third-level it support: challenges, models and case study. International Journal of Services Operations and Informatics 3(3), 242–257 (2008)
7. Spall, J.: Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE Transactions on Automatic Control 37(3), 332–341 (1992)
8. Bhatnagar, S., Hemachandra, N., Mishra, V.: Adaptive multivariate three-timescale stochastic approximation algorithms for simulation based optimization. ACM Transactions on Modeling and Computer Simulation (TOMACS) 21(3) (2011)
9. Borkar, V.S.: Stochastic approximation: a dynamical systems viewpoint. Cambridge Univ. Pr. (2008)