Chapter 1
Introduction

1.1 Introduction

Optimization methods play an important role in many disciplines such as signal processing, communication networks, neural networks, economics, operations research, manufacturing systems, vehicular traffic control, service systems and several others. For instance, in a general communication network, a goal could be to optimally allocate link bandwidth amongst competing traffic flows. Similarly, an important problem in the setting of traffic signal control is to dynamically find the optimal order to switch traffic lights at signal junctions and the amount of time that a lane signal should be green when inputs such as the number of vehicles waiting at other lanes are provided. In the case of a manufacturing plant, an important problem is to decide the optimal order in which to allocate machine capacity for manufacturing various products on any day given the demand patterns for various products. These are only a few specific instances of innumerable problems across various disciplines that fall within the broad category of optimization problems. A usual way to model these problems analytically is by defining an objective or a cost function whose optimum constitutes the desired solution. For instance, in the case of the traffic signal control problem, a cost function could be the sum of queue lengths of vehicles waiting across all lanes at a red signal intersection. Thus, an optimal signal switching order would ensure that the sum of the queue lengths of waiting vehicles is minimized and thereby traffic flows are maximized. In general, a cost function is designed to penalize the less desirable outcomes. However, in principle, there can be several cost functions that have the same (or common) desired outcome as their optimum point. Suitably designing a cost objective in order to obtain the desired outcome in a reasonable amount of time when following a computational procedure could be a domain-specific problem. For instance, in the context of the traffic signal control problem mentioned above, another cost objective with the same optimum could be the sum of squared queue lengths of waiting vehicles instead of the sum of queue lengths. Optimization problems can be deterministic or stochastic, as well as they can be static or dynamic. We discuss this issue in more detail below.

S. Bhatnagar et al.: Stochastic Recursive Algorithms for Optimization, LNCIS 434, pp. 3–12.
springerlink.com

A general optimization problem that we shall be concerned about for the most part in this book has the following form:

$$\text{Find } \theta^* \text{ that solves } \min_{\theta \in C} J(\theta), \tag{1.1}$$

where $J : \mathscr{R}^N \to \mathscr{R}$ is called the objective function, θ is a tunable N-dimensional parameter and $C \subset \mathscr{R}^N$ is the set in which θ takes values. If one has complete information about the function J and its first and higher order derivatives, and about the set C, then (1.1) is a deterministic optimization problem. If on the other hand, J is obtained as $J(\theta) = E_{\xi}[h(\theta, \xi)]$, where $E_{\xi}[\cdot]$ is the expected value over noisy observations or samples $h(\theta, \xi)$ of the cost function with *random noise* ξ, and one is allowed to observe only these samples (without really knowing J), then one is in the realm of stochastic optimization. Such problems are more challenging because of the added complexity of not knowing the cost objective $J(\cdot)$ precisely and to find the optimum parameter only on the basis of the aforementioned noisy observations.

As we shall subsequently see, many times one resorts to search algorithms in order to find an optimum point, i.e., a solution to (1.1). In stochastic optimization algorithms, it is not uncommon to make a random choice in the search direction – in fact most of our treatment will be centered around such algorithms. Thus, a second distinction between deterministic and stochastic optimization problems lies in the way in which search progresses - a random search algorithm invariably results in the optimization setting being stochastic as well.

Suppose now that the objective function J has a *multi-stage* character, i.e., is of the form $J(\theta) = \sum_{i=1}^{N} E[h_i(X_i)]$, where N denotes the number of stages and X_i is the state of an underlying process in stage i, $i = 1, \ldots, N$. The state captures the most important attributes of the system that are relevant for the optimization problem. Further, h_i denotes a stage and state-dependent cost function. Let $\theta \triangleq (\theta_1, \ldots, \theta_N)^T$ denote a vector of parameters θ_j, $j = 1, \ldots, N$ and let X_i depend on θ. The idea here is that optimization can be done one stage at a time over N stages after observing the state X_i in each stage i. Here, the value θ_i of the parameter in stage i has a bearing on the cost of all subsequent stages $i+1, \ldots, N$. This in short is the problem of dynamic optimization. Approaches such as dynamic programming are often used to solve dynamic optimization problems. Other manifestations of dynamic optimization, say over an infinite number of stages or in continuous time also exist. In relation to the above (multi-stage) problem, in static optimization, one would typically perform a single-shot optimization where the parameters $\theta_1, \ldots, \theta_N$ would be optimized all at once in the first stage itself. Broadly speaking while in a dynamic optimization problem with multiple stages, one makes decisions instantly as states are revealed, in static optimization, there is no explicit notion of time or perhaps even state as all decisions can be made at once.

An important class of multi-stage problems are those with an infinite number of stages and where the objective function is a long-run average over single-stage cost functions. More precisely, the objective function in this case has the form

$$J(\theta) = \lim_{N\to\infty} \frac{1}{N} E\left[\sum_{i=1}^{N} h_i(X_i)\right], \tag{1.2}$$

where X_i as before is the state in stage i that we assume depends on the parameter θ. An objective as (1.2) would in most cases not be analytically known. A usual search procedure to find the optimum parameter in such problems would run into the difficulty of having to estimate the cost over an infinitely long trajectory before updating the parameter estimate, thereby making the entire procedure very tedious.

Another important class of optimization problems is that of constrained optimization. Here, the idea is to optimize a given objective or cost function subject to constraints on the values of additional cost functions. Thus consider the following variation to the basic problem (1.1).

$$\text{Find } \theta^* \text{ for which } J(\theta^*) = \min_{\theta\in C}\{J(\theta) \mid G_i(\theta) \le \alpha_i,\ i = 1,\ldots,p\}. \tag{1.3}$$

Here, $G_i(\cdot)$ and α_i, $i = 1,\ldots,p$ are certain additional cost functions and constants, respectively, that constitute the functional constraints. In the context of the traffic signal control problem where the objective function to be minimized is the sum of queue lengths on the various lanes, constraints could be put for the traffic on the side roads so that the main road traffic gets higher priority. For instance, a constraint there could specify that the traffic signal for a side road lane can be switched to green only provided the number of vehicles waiting on such a lane exceeds ten. Similarly, in a communication network, the objective could be to maximize the average throughput. A constraint there could specify that the average delay must be below a threshold. Another constraint could similarly be on the probability of packet loss during transmission being below a small constant, say 0.01.

While for the most part, we shall be concerned with optimization problems of the form (1.1), we shall subsequently also consider constrained optimization problems of the type (1.3). The objective function (and also the constraint functions in the case of (1.3)) will be considered to be certain long-run average cost functions.

We shall present various stochastic recursive search algorithms for these problems. Many of the stochastic search algorithms for optimization can be viewed as stochastic (i.e., with noise) counterparts of corresponding deterministic search algorithms such as gradient and Newton methods. In the setting of stochastic optimization, where the form of the objective function as well as its derivatives is unknown, one needs to resort to estimation of quantities such as the gradient and Hessian from noisy function measurements or else through simulation. A finite difference estimate of the gradient as proposed by Kiefer and Wolfowitz [18] requires a number of function measurements or simulations that is linear in the number of parameter components. A similar estimate of the Hessian [14] requires a number of function measurements that is quadratic in the number of measurements or simulations. When the parameter dimension is large, algorithms with gradient/Hessian estimators as above would be computationally inefficient because such algorithms would update once only after all the required function measurements have been made or simulations conducted. It is here that *simultaneous perturbation methods* play a significant role. In a paper published in 1992, Spall presented the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm that estimated the gradient of the objective function using exactly two function measurements (or simulations) made from perturbed values of the parameter, where each component of the parameter is perturbed along random directions using independent random variates most commonly distributed according to the Bernoulli distribution. A second well-known simultaneous perturbation technique that in fact came before SPSA was the smoothed functional (SF) scheme [17]. The idea in this scheme is some what similar to SPSA, however, the form of the gradient estimator is considerably different as perturbations that are distributed as per the Gaussian, Cauchy or uniform distributions can be used. A basic format for the simultaneous perturbation technique is described in Fig. 1.1.

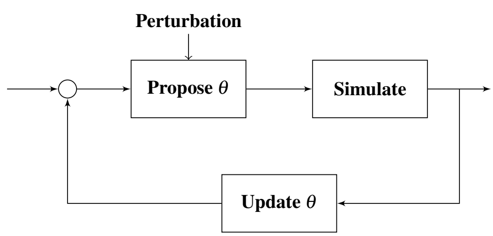

Fig. 1.1 Overall flow of a basic simultaneous perturbation algorithm.

During the course of the last ten to fifteen years, there has been a spurt of activity in developing Newton-based simultaneous perturbation methods. In [27] and [3], Newton-based analogs of the SPSA method were proposed. Further, in [4], Newton-based analogs of the SF algorithm have been proposed. We may mention here that in this text, by *simultaneous perturbation methods*, we refer to the entire family of algorithms that are based on either gradient or gradient and Hessian estimates that are obtained using some form of simultaneous random perturbations. While for the most part, we shall be concerned with static optimization problems, we shall also consider later, the problem of dynamic stochastic control or of decision making under uncertainty over a sequence of time instants. This problem will subsequently be cast as one of dynamic parameter optimization. We shall also present towards the end, applications of the proposed methods and algorithms to service systems, road traffic control and communication networks. A common unifying thread in most of the material presented in this text is of simultaneous perturbation methods.

1.2 Overview of the Remaining Chapters

We now provide a brief overview of the remainder of this book. In Chapter 2, we briefly discuss well-known local search algorithms. These have been described mainly for the case of deterministic optimization. However, we also discuss briefly the case of stochastic optimization as well. The algorithms for stochastic optimization that we present in later chapters will be based on these algorithms.

The fundamental stochastic algorithm due to Robbins and Monro [22] is almost six decades old. It estimates the zeros of a given objective function from noisy cost samples. Most stochastic search algorithms can be viewed as variants of this algorithm. In Chapter 3, we describe the R-M algorithm. We also present in this chapter, a general multi-timescale stochastic approximation algorithm that can be viewed as a variant of the R-M algorithm. Multi-timescale stochastic approximation algorithms play a significant role in the case of problems where the computational procedure would typically involve two nested loops where an outer loop update can happen only upon convergence of the inner loop procedure. A specific instance is the case when the objective function is a long-run average cost of the form (1.2). Such an objective function is useful in scenarios where one is interested in optimizing steady-state system performance measures, such as minimizing long-run average delays in a vehicular traffic network or the steady-state loss probability in packet transmissions in a communication network. A regular computational procedure in this case would perform the outer loop (parameter) update only after convergence of the inner loop procedure (viz., after obtaining the long-run average cost corresponding to a given parameter update). The same effect can be obtained with the use of coupled simultaneous stochastic updates that are however governed with diminishing step-size schedules that have different rates of convergence - the faster update governed with a slowly diminishing schedule and vice versa. Borkar [12, 13] has given a general analysis of these algorithms. We discuss the convergence of both the R-M and the multi-timescale algorithms.

Amongst the first stochastic gradient search algorithms based on estimating the gradient of the objective function using noisy cost samples is the Kiefer-Wolfowitz (K-W) algorithm [18] due to Kiefer and Wolfowitz. We review this algorithm in Chapter 4. While it was originally presented for the case of scalar parameters, in the case of vector-valued parameters, the K-W algorithm makes function measurements after perturbing at most one parameter component. Thus, K-W is not efficient under high-dimensional parameters since the number of function measurements or system simulations required to estimate the gradient grows linearly with the parameter dimension.

Spall invented the simultaneous perturbation stochastic approximation (SPSA) algorithm [23], [28] that requires only two function measurements at each instant regardless of the parameter dimension, by simultaneously perturbing all parameter components using a class of i.i.d. random variables. The most commonly used perturbations in this class are symmetric, ± 1-valued, Bernoulli-distributed random variables. A one-simulation version of this algorithm was subsequently presented in [24]. However, it was not found to be as effective as regular two-simulation SPSA. In [7], certain deterministic constructions for the perturbation random variables have been explored for both two-simulation and one-simulation SPSA. These have been found to yield better results as compared to their random perturbation counterparts. We review the SPSA algorithm and its variants in Chapter 5.

Katkovnik and Kulchitsky [17] presented a smoothed functional (SF) approach that is another technique to estimate the gradient of the objective function using random perturbations. This technique is some what different from SPSA. In particular, the properties required of the perturbation random variables here are seen to be most commonly satisfied by Gaussian and Cauchy distributed random variables. If one considers a convolution of the gradient of the objective function with a smoothing density function (such as that of Gaussian or Cauchy random variables), then through a suitable integration-by-parts argument, one can rewrite the same as a convolution of the gradient of the probability density function (p.d.f.) with the objective function itself. The derivative of the smoothing p.d.f. is seen to be a scaled version of the same p.d.f. This suggests that if the perturbations are generated using such p.d.fs, only one function measurement or system simulation is sufficient to estimate the gradient of the objective (in fact, the convolution of the gradient, that however converges to the gradient itself in the scaling limit of the perturbation parameter). A two-simulation variant of this algorithm that incorporates balanced estimates has been proposed in [29] and found to perform better than its one-simulation counterpart. We review developments in the gradient-based SF algorithms in Chapter 6.

Spall [27] presented simultaneous perturbation estimates for the Hessian that incorporate two independent perturbation sequences that are in the same class of sequences as used in the SPSA algorithm. The Hessian estimate there is based on four function measurements or system simulations, two of which are the same as those used for estimating the gradient of the objective. In [3], three other Hessian estimators were proposed. These are based on three, two and one system simulation(s), respectively. In Chapter 7, we review the simultaneous perturbation estimators of the Hessian. An issue with Newton-based algorithms that incorporate the Hessian is in estimating the inverse of the Hessian matrix at each update epoch. We also discuss in this chapter some of the recent approaches for inverting the Hessian matrix.

Bhatnagar [4] developed two SF estimators for the Hessian based on one and two system simulations, respectively, when Gaussian p.d.f. is used as the smoothing function. Using an integration-by-parts argument (cf. Chapter 6), twice, the Hessian estimate is seen to be obtained from a single system simulation itself. A two-sided balanced Hessian estimator is, however, seen to perform better than its one-sided counterpart. An interesting observation here is that both the gradient and the Hessian estimates are obtained using the same simulation(s). We review the SF estimators of the Hessian matrix in Chapter 8.

In Chapter 9, we consider the case when the optimization problem has a form similar to (1.1); however, the underlying set C is discrete-valued. Further, we shall let the objective function be a long-run average cost as with (1.2). In [11], two gradient search algorithms based on SPSA and SF have been proposed for this problem. A randomized projection approach was proposed there that is seen to help in adapting the continuous optimization algorithms to the discrete setting. We present another approach based on certain generalized projections that can be seen to be a mix of deterministic and randomized projection approaches, and result in the desired smoothing of the dynamics of the underlying process. Such a projection mechanism would also result in a lower computational complexity as opposed to a fully randomized projection scheme.

Next, in Chapter 10, we will be concerned with constrained optimization problems with similar objective as (1.3). We shall, in particular, be concerned here with the case when the objective has a long-run-average form similar to (1.2). Thus, in such cases, neither the objective nor the constraint region is known analytically to begin with. In [8], stochastic approximation algorithms based on SPSA and SF estimators for both the gradient and the Hessian have been presented. The general approach followed is based on forming the Lagrangian – the Lagrange multipliers are updated on a slower timescale than the parameter that, in turn, is updated on a slower scale in comparison to that on which data gets averaged. We will review these algorithms in Chapter 10.

Reinforcement learning (RL) algorithms [2] are geared towards solving stochastic control problems using only real or simulated data when the system model (in terms of the transition probabilities) is not known. Markov decision process (MDP) is a general framework for studying such problems. Classical approaches such as policy iteration and value iteration for solving MDP require knowledge of transition probabilities. Many RL algorithms are stochastic recursive procedures aimed at solving such problems when transition probabilities are unknown. Actor-critic (AC) algorithms are a class of RL algorithms that are based on policy iteration and involve two loops - the outer loop update does policy improvement while the inner loop procedure is concerned with policy evaluation. These algorithms thus incorporate two-timescale stochastic approximation. In [10, 1, 6], AC algorithms for various cost criteria such as infinite horizon discounted cost, long-run average cost as well as total expected finite horizon cost, that incorporate simultaneous perturbation gradient estimates have been proposed. We shall review the development of the infinite horizon algorithms in Chapter 11.

Chapter 12 considers the problem of optimizing staffing levels in service systems. The aim is to adapt the staffing levels as they are labor intensive and have a time varying workload. This problem is, however, nontrivial due to a large number of parameters and operational variations. Further, any staffing solution is constrained to maintain the system in steady-state and be compliant to aggregate SLA constraints. We formulate the problem using the constrained optimization framework where the objective is to minimize the labor cost in the long run average sense and the constraint functions are long run averages of the SLA and queue stability constraints. Using the ideas of the algorithms proposed in Chapter 10 for a generalized constrained optimization setting, we describe several simulation optimization methods that have been originally proposed in [19] for solving the labor cost optimization problem. The presented algorithms are based on SPSA and SF gradient/Hessian estimates. These algorithms have been seen in [19] to exhibit better overall performance vis-a-vis the state-of-the-art optimization tool-kit OptQuest, while being more than an order of magnitude faster than Optquest.

In Chapter 13, we consider the problem of finding optimal timings and the order in which to switch traffic lights given dynamically evolving traffic conditions. We describe here applications of the reinforcement learning and stochastic optimization approaches in order to maximize traffic flow through the adaptive control of traffic lights. We assume, however, as in the case of real-life situations that only rough estimates of the congestion levels are available, for instance, whether congestion is below a lower threshold, above an upper threshold or is in between the two. All our algorithms incorporate such threshold levels in the feedback policies and find optimal policies given a particular set of thresholds. For instance, in a recent work [21], we considered Q-learning-based traffic light control (TLC) where the features are obtained using such (aforementioned) thresholds. We also describe similar other algorithms based on simulation optimization methods. An important question then is to find optimal settings for the thresholds themselves. We address this question by incorporating simultaneous perturbation estimates to run in tandem with other algorithms. An important observation is that our algorithm shows significantly better empirical performance as compared to other related algorithms in the literature. Another interesting consequence of our approach is that when applied together with reinforcement learning algorithms, such methods result in obtaining an *optimal* set of features from a given parametrized feature class.

In Chapter 14, we select and discuss three important problems in communication networks, where simultaneous perturbation approaches have been found to be significantly useful. We first consider the problem of adaptively tuning the parameters in the case of random early detection (RED) adaptive queue management scheme proposed for TCP/IP networks. The original scheme proposed by Floyd [15] considers a fixed set of parameters regardless of the network and traffic conditions. We address this problem using techniques from constrained optimization [20] and apply simultaneous perturbation approaches that are found to exhibit excellent performance. Next, we consider the problem of tuning the retransmission probability parameter for the slotted Aloha multi-access communication system. The protocol as such specifies a fixed value for the same regardless of the number of users sending packets on the channel and the channel conditions. We propose a stochastic differential equation (SDE)-based formulation [16, 9] in order to find an optimal parameter trajectory over a finite time horizon. We also consider the problem of optimal pricing in the Internet. The idea here is that in order to provide a higher quality of service to a user who is willing to pay more, one needs to find optimal strategies for fixing prices of the various services offered. Our techniques [30] play a role here as well and are found to exhibit significantly better performance in comparison to other known methods.

Finally, in Appendices A-E, we present some of the basic material needed in the earlier chapters. In particular, we present (a) convergence notions for a sequence of random vectors, (b) results on martingales and their convergence, (c) ordinary differential equations, (d) the Borkar and Meyn stability result, and (e) the Kushner-Clark theorem for convergence of projected stochastic approximations. Some of the background material as well as the main results used in other chapters have also been summarized in these appendices.

1.3 Concluding Remarks

Stochastic approximation algorithms are one of the most important class of techniques for solving optimization problems involving uncertainty. Simultaneous perturbation approaches for optimization have evolved into a rich area by themselves from the viewpoint of both theory and numerous highly successful applications. Several estimators for the gradient and Hessian that involve simultaneous perturbation estimates have been developed in recent times that are seen to show excellent performance. SPSA and SF algorithms constitute powerful methods for stochastic optimization that have been found useful in many disciplines of science and engineering. The book reference of [28] provides an excellent account of SPSA. Surveys on the SPSA algorithm are available in [26], [25]. Also, [5] provides a more recent survey on simultaneous perturbation algorithms involving both SPSA and SF estimators. The current text is a significantly expanded version of [5].

References

1. Abdulla, M.S., Bhatnagar, S.: Reinforcement learning based algorithms for average cost Markov decision processes. Discrete Event Dynamic Systems 17(1), 23–52 (2007)
2. Bertsekas, D.P., Tsitsiklis, J.N.: Neuro-Dynamic Programming. Athena Scientific, Belmont (1996)
3. Bhatnagar, S.: Adaptive multivariate three-timescale stochastic approximation algorithms for simulation based optimization. ACM Transactions on Modeling and Computer Simulation 15(1), 74–107 (2005)
4. Bhatnagar, S.: Adaptive Newton-based smoothed functional algorithms for simulation optimization. ACM Transactions on Modeling and Computer Simulation 18(1), 2:1–2:35 (2007)
5. Bhatnagar, S.: Simultaneous perturbation and finite difference methods. Wiley Encyclopedia of Operations Research and Management Science 7, 4969–4991 (2011)
6. Bhatnagar, S., Abdulla, M.S.: Simulation-based optimization algorithms for finite horizon Markov decision processes. Simulation 84(12), 577–600 (2008)
7. Bhatnagar, S., Fu, M.C., Marcus, S.I., Wang, I.J.: Two-timescale simultaneous perturbation stochastic approximation using deterministic perturbation sequences. ACM Transactions on Modelling and Computer Simulation 13(2), 180–209 (2003)
8. Bhatnagar, S., Hemachandra, N., Mishra, V.: Stochastic approximation algorithms for constrained optimization via simulation. ACM Transactions on Modeling and Computer Simulation 21, 15:1–15:22 (2011)
9. Bhatnagar, S., Karmeshu, Mishra, V.: Optimal parameter trajectory estimation in parameterized sdes: an algorithmic procedure. ACM Transactions on Modeling and Computer Simulation (TOMACS) 19(2), 8 (2009)
10. Bhatnagar, S., Kumar, S.: A simultaneous perturbation stochastic approximation based actor-critic algorithm for Markov decision processes. IEEE Transactions on Automatic Control 49(4), 592–598 (2004)
11. Bhatnagar, S., Mishra, V., Hemachandra, N.: Stochastic algorithms for discrete parameter simulation optimization. IEEE Transactions on Automation Science and Engineering 9(4), 780–793 (2011)

12. Borkar, V.S.: Stochastic approximation with two timescales. Systems and Control Letters 29, 291–294 (1997)
13. Borkar, V.S.: Stochastic Approximation: A Dynamical Systems Viewpoint. Cambridge University Press and Hindustan Book Agency (Jointly Published), Cambridge and New Delhi (2008)
14. Fabian, V.: Stochastic approximation. In: Rustagi, J.J. (ed.) Optimizing Methods in Statistics, pp. 439–470. Academic Press, New York (1971)
15. Floyd, S., Jacobson, V.: Random early detection gateways for congestion avoidance. IEEE/ACM Transactions on Networking 1(4), 397–413 (1993)
16. Karmeshu, Bhatnagar, S., Mishra, V.: An optimized sde model for slotted aloha. IEEE Transactions on Communications 59(6), 1502–1508 (2011)
17. Katkovnik, V.Y., Kulchitsky, Y.: Convergence of a class of random search algorithms. Automation Remote Control 8, 1321–1326 (1972)
18. Kiefer, E., Wolfowitz, J.: Stochastic estimation of the maximum of a regression function. Ann. Math. Statist. 23, 462–466 (1952)
19. Prashanth, L.A., Prasad, H., Desai, N., Bhatnagar, S., Dasgupta, G.: Simultaneous perturbation methods for adaptive labor staffing in service systems. Tech. rep., Stochastic Systems Lab, IISc (2012), http://stochastic.csa.iisc.ernet.in/www/research/files/IISc-CSA-SSL-TR-2011-4-rev2.pdf
20. Patro, R.K., Bhatnagar, S.: A probabilistic constrained nonlinear optimization framework to optimize RED parameters. Performance Evaluation 66(2), 81–104 (2009)
21. Prashanth, L., Bhatnagar, S.: Reinforcement learning with function approximation for traffic signal control. IEEE Transactions on Intelligent Transportation Systems 12(2), 412–421 (2011)
22. Robbins, H., Monro, S.: A stochastic approximation method. Ann. Math. Statist. 22, 400–407 (1951)
23. Spall, J.C.: Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE Trans. Auto. Cont. 37(3), 332–341 (1992)
24. Spall, J.C.: A one-measurement form of simultaneous perturbation stochastic approximation. Automatica 33, 109–112 (1997)
25. Spall, J.C.: An overview of the simultaneous perturbation method for efficient optimization. Johns Hopkins APL Technical Digest 19, 482–492 (1998)
26. Spall, J.C.: Stochastic optimization, stochastic approximation and simulated annealing. In: Webster, J.G. (ed.) Wiley Encyclopedia of Electrical and Electronics Engineering, vol. 20, pp. 529–542. John Wiley and Sons, New York (1999)
27. Spall, J.C.: Adaptive stochastic approximation by the simultaneous perturbation method. IEEE Trans. Autom. Contr. 45, 1839–1853 (2000)
28. Spall, J.C.: Introduction to Stochastic Search and Optimization. John Wiley and Sons, New York (2003)
29. Styblinski, M.A., Tang, T.S.: Experiments in nonconvex optimization: stochastic approximation with function smoothing and simulated annealing. Neural Networks 3, 467–483 (1990)
30. Vemu, K.R., Bhatnagar, S., Hemachandra, N.: Optimal multi-layered congestion based pricing schemes for enhanced qos. Computer Networks (2011), http://dx.doi.org/10.1016/j.comnet.2011.12.004