IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 5, 2021, accepted July 27, 2021, date of publication August 9, 2021, date of current version August 19, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3103767

Multi-Time Scale Smoothed Functional With Nesterov's Acceleration

ABHINAV SHARMA[1], K. LAKSHMANAN[2], RUCHIR GUPTA[3], (Senior Member, IEEE), AND ATUL GUPTA[1]

[1]Department of Computer Science, PDPM Indian Institute of Information Technology at Jabalpur, Jabalpur, Madhya Pradesh 482005, India
[2]Department of Computer Science, IIT Banaras Hindu University (BHU) at Varanasi, Varanasi, Uttar Pradesh 221005, India
[3]Department of Computer Science, Jawaharlal Nehru University (JNU), Delhi 110067, India

Corresponding author: Ruchir Gupta (rgupta.cse@itbhu.ac.in)

This work was supported in part by Nokia Solutions and Networks, Bengaluru, India.

ABSTRACT Smoothed functional (SF) algorithm estimates the gradient of the stochastic optimization problem by convolution with a smoothening kernel. This process helps the algorithm to converge to a global minimum or a point close to it. We study a two-time scale SF based gradient search algorithm with Nesterov's acceleration for stochastic optimization problems. The main contribution of our work is to prove the convergence of this algorithm using the stochastic approximation theory. We propose a novel Lyapunov function to show the associated second-order ordinary differential equations' (o.d.e.) stability for a non-autonomous system. We compare our algorithm with other smoothed functional algorithms such as Quasi-Newton SF, Gradient SF and Jacobi Variant of Newton SF on two different optimization problems: first, on a simple stochastic function minimization problem, and second, on the problem of optimal routing in a queueing network. Additionally, we compared the algorithms on real weather data in a weather prediction task. Experimental results show that our algorithm performs significantly better than these baseline algorithms.

INDEX TERMS Multi-Stage queueing networks, Nesterov's acceleration, simulation, smoothed functional algorithm, stochastic approximation algorithms, stochastic optimization.

I. INTRODUCTION

Optimization problems deal with minimizing (or maximizing) the value of an objective function [1]. When parameters of the objective function or the optimization algorithm have randomness, then the process of optimization is termed as Stochastic Optimization (SO) [2]. It has applications in various fields such as machine learning [3], finance, supply chain [4], network optimization [5] and optimization with information uncertainty [6], [7]. These algorithms generally involve the estimation of the gradient of the objective. One of the most popular algorithm in this regard is Stochastic Gradient Descent (SGD) [8]. It was evolved from the works of Robbins and Monro [9], and it estimates the gradients of the cost function. If the objective function is represented by $J(\theta)$, where $\theta \in \mathbb{R}^d$ is a parameter, then SGD utilizes gradient of objective function ($\nabla_\theta J(\theta)$) to update the parameter in opposite direction of the gradient. However, in its vanilla form, SGD suffers from slower convergence on large data [10].

An improvement to this basic algorithm is the classical momentum [11] based algorithm where along with negative gradient, a decreasing weighted sum of past updates is also utilized. Due to the inclusion of previous updates, the latest update tend to move faster if it is in the same direction. This phenomenon is applicable even if the current gradient value is small (i.e., when the function's curvature is low) [12]. However, this method is not very efficient when the direction of the gradient oscillates (i.e., when the function has a "deep valley"). Nesterov proposed an accelerated gradient descent technique to overcome this problem [13].

In Nesterov's Accelerated Gradient (NAG), the update is done in two parts. (1) A partial step is taken in the same direction as the previous momentum vector. (2) A second partial step is taken in the current gradient's direction to accelerate the gradient search. However, this direction is reversed if the first partial step overshoots the minima, which decreases the overall update value. It contrasts with classical momentum-based methods as they oscillate in such scenarios [14]. NAG is discussed in detail in Section IV.

The associate editor coordinating the review of this manuscript and approving it for publication was Mauro Gaggero.

In many scenarios, calculating the gradient is computationally expensive [15]. Thus, calculating the gradient directly may not always be possible, but it can be estimated from a simulation. Kiefer and Wolfowitz [16] employed a direct gradient estimation technique in a general setting. The number of parameter values that need to be simulated in their work is at least as large as the search space dimension. Later, SPSA [17] was proposed, which requires only two function measurements for estimating the gradient regardless of the optimization problem's dimension, thus using much less computational effort. It is based on gradient estimation using independent and identically distributed randomly perturbed parameters.

Another related simultaneous perturbation technique is smoothed functional (SF) scheme. In this, the objective function's gradient is convoluted with a multivariate Gaussian distribution, resulting in smoothed gradient updates [18]. This convolution step slightly changes the absolute minima, but the smoothed gradient surface [19] compensates this effect by reducing the total number of local minima as they get disappeared due to smoothening. In addition, the number of simulations per iteration in SF algorithms are independent of the dimension of the objective function. The required samples are collected from the simulation of the objective function. The one-simulation method was proposed by [20], where samples from a single-simulation are enough for each gradient estimate. However, Bhatnagar *et al.* [18], and Styblinski *et al.* [21], showed that two-simulation methods[1] outperforms SF based single-simulation technique. It requires only two simulations to compute the gradient irrespective of the dimension, unlike direct estimation. Thus, the SF method reduces the number of simulations. Later, the two-simulation SF technique is combined with a Quasi-Newton update (BFGS) [22] for both constrained and unconstrained SO. Quasi-Newton is an improvement over regular Newton algorithms as the former only approximates the Hessian inverse, unlike the latter, which computes it.

Thus, it is clear that SF algorithms help in avoiding many local minima's by smoothening the objective function's gradient. Moreover, a momentum-based descent algorithm accelerates the gradient vector in the right direction in the search space, leading to less computational time, and it also overcomes the oscillation of noisy gradients. To the best of our knowledge, momentum-based gradient descent algorithms have not been considered with multi-time scale [23] (recursions involves more than one step size parameters of different values) smoothed functional techniques. This motivates us to explore these algorithms. Thus, the novelty of our work is the use of Nesterov's Accelerated Gradient Descent (NAG) with SF technique (NAG-SF algorithm) in a multi-time scale fashion to minimize the average cost objective.

The proposed algorithm aims to speed up the slow convergence of multi-time scale stochastic optimization algorithms. Though multi-time scale algorithms efficiently use different step-sizes to avoid nested loops, increasing computation speed. However, we consider acceleration methods together with this to improve the rate of convergence also. We have proved the convergence of our method by analyzing the associated ODE. Experimentally our algorithm is seen to perform better than other variants of SF algorithms. We have tested our algorithm on a routing problem that mimics a simplified model of the internet. It is a multi-stage shortest path problem, similar to which are considered in [22], [24]–[26]. Next, the performance is tested on a simple stochastic function minimization problem and compared with non-accelerated versions of smoothed functional algorithms such as Quasi-Newton SF [22], Gradient SF and Jacobi Variant of Newton SF [18]. Moreover, we tested the algorithms on a weather classification problem using real weather data from the National Center for Atmospheric Research (NCAR) [27] for a period of 365 days between year 2008 and 2009.

The main contribution of our work is:

- Novel SF based Nesterov's accelerated algorithm for stochastic convex optimization.
- Analyze the associated second-order o.d.e. of the system while Bhatnagar's work [11] uses a first-order o.d.e.
- Introduction of a novel Liapunov function to analyze the o.d.e.
- Use of stability theorem (Theorem 5) to analyze the stability of the non-autonomous o.d.e.

Although the proof structure for the convergence of NAG-SF is similar to Bhatnagar's work [18], the proof is indeed different. We introduce a stability theorem for non-autonomous o.d.e. Experimental analysis suggests that our work is better than the previous research.

The rest of the paper is organized as follows. Section II provides the problem definition for the optimization setting. Section III describes the background of the smoothed functional algorithms in general, two-time scale scheme and the basic structure of Nesterov's momentum algorithm. The proposed NAG-SF algorithm is given in Section IV. A mathematical analysis and convergence proof of the proposed algorithm is discussed in Section V. Section VI provides the experimental setup and results to compare against other baselines. Conclusions are drawn in Section VII.

II. PROBLEM SETTING

We let $\{X_n, n \geq 1\}$ be an $\mathbb{R}^d$-valued parameterized ergodic Markov process, that takes values in a non-empty compact and convex set $C \subset \mathbb{R}^K$. With n being time step, let, $\theta(n) \in C$ be a tunable parameter for the transition kernel of the process $\{X_n\}$. Thus, at a given instant $n+1$, the transition kernel uses $\theta(n)$ and X_n to generate the process X_{n+1}.

[1] One-simulation methods require single objective function measurement per iteration, and two-simulation methods require two measurements of objective function per iteration.

Let a cost function $h : \mathbb{R}^d \to \mathbb{R}^+$ be defined on process X_n, such that when the process is in state x, then, $h(x)$ be the single-stage cost incurred. The long-run average cost $J(\cdot)$ of this cost function h is defined as:

$$J(\theta) \triangleq \lim_{l\to\infty} \frac{1}{l} \sum_{j=0}^{l-1} h(X_j). \tag{1}$$

The objective is to minimize this long-run average cost $J(\cdot)$ for all $\theta \in C$. The given limit exists due to the ergodicity of process X_n for all $\theta \in C$. We propose an iterative algorithm for this optimization problem.

Next, assume that, $\{\theta(n)\}$ be a sequence of random parameters obtained using an iterative scheme on which the process $\{X_n\}$ depends and let $\mathcal{H}_n = \sigma(\theta(m), X_m, m \leq n)$, $n \geq 1$ be the sequence of associated $\sigma-$fields. We refer $\{\theta(n)\}$ to be non-anticipative, if, for all Borel sets $A \subset \mathbb{R}^d$,

$$P(X_{n+1} \in A|\mathcal{H}_n) = p(\theta(n), X_n, A),$$

where $p(\cdot)$ denotes the transition kernel. In particular, the joint process $\{(X_n, \theta(n))\}$ is Markovian under a non-anticipative $\{\theta(n)\}$.

We make the following assumptions for the analysis of NAG-SF algorithm:

Assumption 1: $J(\theta)$ is a twice continuously differentiable function in θ with bounded third derivatives.

Assumption 2: The single stage cost $h(\cdot)$ is a Lipschitz continuous function, that is, $\exists\Lambda$ which is a real positive constant, such that $\forall k_1, k_2 \in$ domain of h, $|h(k_1)-h(k_2)| \leq \Lambda|k_1 - k_2|$.

Assumption 3: There exist $\epsilon_0 > 0$, $\mathcal{K} \subset \mathbb{R}^d$ compact and $V \in C(\mathbb{R}^d)$ such that $\lim_{\|x\|\to\infty} V(x) = \infty$ and under any non-anticipative $\{\theta(n)\}$,

1) $\sup_n \mathbb{E}[V(X_n)^2] < \infty$ *and*
2) $\mathbb{E}[V(X_{n+1})|\mathcal{H}_n] \leq \epsilon_0$, *whenever* $X_n \notin \mathcal{K}$, $n \geq 0$.

Assumption 1 and 2 are standard requirements [28]. In particular, Assumption 1 ensures that the Hessian of the objective function exists. Assumption 3 is needed for the existence of a stochastic Lyapunov function. It ensures the stability of the $\mathbb{R}^d$-valued Markov process under a tunable parameter, as the cost function in a state variable is taken as Lipschitz continuous. It also ensures that the Markov process has uniformly bounded finite moments [28].

Assumption 3 is required for the stability of the system with $\mathbb{R}^d$-valued Markov process. It ensures a stable system under a tunable parameter since, in the proposed setting, the cost function $h(\cdot)$ is not bounded. As a consequence of Assumption 3, all finite moments of the Markov process remain uniformly bounded. We let $\|\cdot\|$ denote the Euclidean norm. The objective here is to find the local minimum using an iterative algorithm. Hence, it is required that the Hessian estimate remains positive definite and symmetric after each iteration. For this, we project the Hessian estimate to the space of positive definite and symmetric matrices using the operator P described previously.

III. BACKGROUND

In 1951, Robbins and Monro [9] proposed a scheme for solving a non-linear equation $h(\theta) = 0$ given noisy measurements of the function. Considering $\{\mathcal{Z}\}$ to be noise sequence, the iteration given by them is:

$$\theta(n+1) = \theta(n) \ + a(n)[h(\theta(n)) + \mathcal{Z}(n)] \tag{2}$$

where $a(n)$ is the step size. Now, to theoretically analyze these type of stochastic approximation algorithms, one popular approach is to view the iteration as a noisy discretization of a limiting o.d.e. Thus, from the standard 'Euler scheme', the corresponding o.d.e. for (2) would be

$$\dot{\theta}(t) = h(\theta(t)) \tag{3}$$

For formal analysis of the stochastic approximation scheme, the following assumptions need to be made:

Assumption 4: Step sizes $\{a(n)\}$ are positive scalars satisfying $\sum_n a(n) = \infty$; $\sum_n a(n)^2 < \infty$

Assumption 5: $\{\mathcal{Z}(n)\}$ is a martingale difference sequence with respect to the increasing family of σ-fields.

$$\mathcal{F}(n) \triangleq \sigma\left(\theta(m), \mathcal{Z}(m), m \leq n\right), n \geq 0$$

Assumption 4 is a standard requirement for the step sizes and Assumption 5 is to define the added noise in the update. Now, a closed set $A \subset \mathbb{R}^d$ is referred to as *invariant set* for the o.d.e. (3) if any trajectory $\theta(n)$, $-\infty < n < \infty$ of (3) remains in set A, ie., it satisfies $\theta(n) \in A \ \forall \ n \in \mathbb{R}$. Furthermore, if for any $\theta, y \in A$ and any $\epsilon > 0, T > 0$, there exist $n \geq 1$ and points $\theta(0) = \theta, \theta(1), \ldots, \theta(n-1), \theta(n) = y$ in A such that the trajectory of (3) initiated at $\theta(i)$ meets with the ϵ-neighbourhood of $\theta(i+1)$ for $0 \leq i < n$ after a time $\geq T$, then it is referred to as *internally chain transitive*.

Under the mentioned assumptions, Benaim [29] gave a convergence result for the update rule mentioned in (2).

Theorem 1: Almost surely, the sequence $\theta(n)$ generated by (2) *converges to a (possibly sample path-dependent) compact connected internally chain transitive invariant set of* (3).

Thus, the iterative scheme given by [9] converge to a compact set. Next, several variations might be present to analyze the stability criteria for the stochastic approximation algorithms, which might be applicable under specific restrictions. These variations might additionally be customized to suit some specific applications. One such variation as given by [30] is as follows: For the o.d.e. described earlier, at any time step T_n, the iterate has to be restricted to a unit ball in $\mathbb{R}^d$ for the trajectory to remain meaningful. Therefore, the iterate is re-scaled back over the time segment $[T_n, T_{n+1})$ when it drifts away. If the original trajectory drifts towards infinity, then there is a corresponding sequence of re-scaled segments. These segments asymptotically track a limiting o.d.e. and are obtained as a scaling limit of 'basic o.d.e.'. The stability condition is met when these segments start drifting towards the origin, which happens when the scaling limit is globally asymptotically stable to the origin. Formally

Assumption 6: The functions $h_u(\theta) \triangleq h(u\theta)/u, u \geq 1$, $\theta \in \mathbb{R}^d$, satisfy $h_u(\theta) \to h_\infty(\theta)$ as $u \to \infty$, uniformly *on compacts for some $h_\infty \in C(\mathbb{R}^d)$. Furthermore, the o.d.e. $\dot{\theta}(t) = h_\infty(\theta(t))$ has the origin as its unique globally asymptotically stable equilibrium.*

Here, the o.d.e. mentioned in assumption (6) is the scaling limit. So now we can state the result from [23].

Theorem 2: Under assumptions (2-6), $\sup_n |\theta(n)| < \infty$ *a.s.*

Thus, the parameter θ remains stable.

Many stochastic approximation algorithm uses multiple time scale approach corresponding to different components of the iteration which induces different time scales into the algorithm. We consider the case of two time scales following [31]. Consider the following iterations

$$\theta(n+1) = \theta(n) + b(n)[h(\theta(n), y(n)) + \mathcal{Z}^{(1)}(n+1)], \quad (4)$$

$$y(n+1) = y(n) + a(n)[g(\theta(n), y(n)) + \mathcal{Z}^{(2)}(n+1)], \quad (5)$$

where $h : \mathbb{R}^{d+k} \to \mathbb{R}^d, g : \mathbb{R}^{d+k} \to \mathbb{R}^k$ are Lipschitz and $\{\mathcal{Z}^{(1)}(n+1)\}, \mathcal{Z}^{(2)}(n+1)$ are martingale difference sequences w.r.t. the increasing σ-fields

$$\mathcal{F}(n) = \sigma(\theta(m), y(m), \mathcal{Z}^{(1)}(m), \mathcal{Z}^{(2)}(m), m \le n), n \ge 0 \quad (6)$$

satisfying

$$\mathbb{E}[\|\mathcal{Z}^{(i)}(n+1)\|^2 | \mathcal{F}(n)] \le K(1 + |\theta(n)|^2 + |y(n)|^2), \ i = 1, 2 \quad (7)$$

for $n \ge 0$. Step sizes $\{a(n)\}, \{b(n)\}$ are positive scalars satisfying

Assumption 7:

$$\sum_n a(n) = \sum_n b(n) = \infty; \sum_n a(n)^2, \sum_n b(n)^2 < \infty, \quad (8)$$

$$a(n) = O(b(n)). \quad (9)$$

The rate of decay is different for different step sizes. If a step size parameter goes to zero faster than the other, the corresponding recursions tend to converge slower. Here, $a(n)$ approaches zero faster than $b(n)$, thus recursions corresponding to $a(n)$ converge slower, though more smoothly than recursions corresponding to step size $b(n)$. Therefore, the time scale governed by $a(n)$ is the slower time scale, and that governed by $b(n)$ is a faster time scale [23].

Considering $\epsilon \to 0$ in limit, the iterations in equation (4) and (5) can be compared with the following o.d.e.

$$\dot{\theta} = h(\theta(t), y(t))/\epsilon \quad (10)$$

$$\dot{y} = g(\theta(t), y(t)) \quad (11)$$

Thus $\theta(\cdot)$ is a fast transient and $y(\cdot)$ the slow component. It then makes sense to think of $y(\cdot)$ as quasi-static (i.e., 'almost a constant') while analyzing the behaviour of $\theta(\cdot)$. This suggests looking at the o.d.e.

$$\dot{\theta}(t) = h(\theta(t), y) \quad (12)$$

where y is held fixed as a constant parameter.

Assumption 8: Equation (12) has a globally asymptotically stable equilibrium $\lambda(y)$ (uniformly in y), where $\lambda : \mathbb{R}^k \to \mathbb{R}^d$ is a Lipschitz map.

Then for sufficiently small values of ϵ we expect $\theta(n)$ to closely track $\lambda(y(n))$ for $n > 0$. In turn this suggests looking at the o.d.e.

$$\dot{y}(t) = g(\lambda(y(t)), y(t)), \quad (13)$$

which should capture the behaviour of $y(\cdot)$ in equation (11) to a good approximation. Suppose that:

Assumption 9: The o.d.e. (13) has a globally asymptotically stable equilibrium y^*

Then we expect $(\theta(n), y(n))$ in (10)–(11) to approximately converge to (i.e., converge to a small neighbourhood of) the point $(\lambda(y^*), y^*)$.

The motivation for analyzing this setup comes from the subsequent considerations. Suppose that an iterative algorithm requires a selected iterative procedure in every iteration. Additionally, that procedure itself is any other iterative algorithm. The conventional approach could be to apply the procedure's output when running it till near-convergence, during every iterate of the outer loop. This is a time-consuming step. However, the aforementioned indicates that we can get a similar impact by running both the inner and outer loops concurrently, albeit on different time scales. Then the inner 'fast' loop sees the outer 'slow' loop as quasi-static, and the latter sees the previous as almost equilibrated. Consider the following stability assumption:

Assumption 10: $\sup_n(\|\theta_n\| + |y_n|) < \infty$*, a.s.*

For the two-time scale approach, the formal convergence analysis is discussed in detail in [19]. For the sake of completeness, the results are as follows:

Lemma 1: Considering assumptions 10, $(x_n, y_n) \to \{(\lambda(y), y : y \in \mathbb{R}^k)\}$ *a.s.*

Using this lemma and Assumption 10, the following result can be obtained [23]:

Theorem 3: $(x_n, y_n) \to (\lambda(y^*), y^*)$ *a.s.*

Thus, this same approach can be used for more time scales. However, it turns out that increasing the time scale has a detrimental effect on the performance of the algorithm [28].

A. SMOOTHED FUNCTIONAL ALGORITHM FOR ESTIMATING STOCHASTIC GRADIENT

Optimizing a general problem can be difficult when many local minima are present. SF algorithms solve this issue by convoluting the objective function's gradient with an operator known as the smoothening kernel (such as Gaussian). Katkovnik and Kulchitsky first proposed this method in [20] where they used a single estimate to approximate the gradient of the objective function with the use of multivariate Gaussian distribution. Later, Bhatnagar *et al.* in [32] presented a two-time scale version (i.e. recursions involving two different step-size schedules) of the one-simulation smoothed functional algorithm. We consider the SF method developed by Bhatnagar in [18], where, a K dimensional multivariate Gaussian density function $G_\beta(\theta - \eta)$ (joint p.d.f. of K independent $N(0, \beta^2)$-distributed random variables) convolute the gradient of the objective function $J(.)$, and is given by:

$$D_{\beta,1}J(\theta) = \int G_\beta(\theta - \eta)\nabla_\eta J(\eta)d\eta \tag{14}$$

where $\beta > 0$ is a scalar parameter which controls the smoothness and $\theta, \eta \in \mathbb{R}^K$ with $\theta \triangleq (\theta_1, \ldots, \theta_K)^T$ and $\eta \triangleq (\eta_1, \ldots, \eta_K)^T$. If the objective function $J(\theta)$ is not well behaved (has fluctuating character), then, the convoluted objective function $D_{\beta,1}J(\theta)$ obtained by the smoothening becomes better behaved, and thus optimization algorithms can provide improved results.

The Gaussian density function $G_\beta(\theta - \eta)$ is defined as:

$$G_\beta(\theta - \eta) = \frac{1}{(2\pi)^{K/2}\beta^K} \exp\left(-\frac{1}{2}\sum_{i=1}^{K} \frac{(\theta_i - \eta_i)^2}{\beta^2}\right), \tag{15}$$

Now, solving equation (14) with equation (15) using integration by parts we get (refer [18])

$$D_{\beta,1}J(\theta) = \mathbb{E}\left[\frac{1}{\beta}\bar{\eta}J(\theta + \beta\bar{\eta}|\theta)\right] \tag{16}$$

where $\bar{\eta} = -\frac{\eta}{\beta}$, and the expectation is w.r.t. another K-dimensional multivariate Gaussian p.d.f. $G(\bar{\eta})$. SF algorithms uses function measurements from simultaneous perturbed parameters which updates the gradient in all component directions. Thus, these algorithms belong to the same class of simultaneous perturbations algorithms like SPSA [17]. The gradient estimator for $J(\theta(n)$ is inspired from Bhatnagar *et al.* [18] and is given by:

$$\nabla J(\theta(n)) = \lim_{\beta\to 0}\lim_{\mathcal{L}\to\infty} \frac{1}{\beta}\frac{1}{\mathcal{L}}\sum_{n=1}^{\mathcal{L}} \bar{\eta}(n)J\left(\theta(n) + \beta\bar{\eta}(n)\right) \tag{17}$$

In a similar fashion, the gradient update for two-simulation gradient estimator can be obtained by:

$$\nabla J(\theta(n)) = \lim_{\beta\to 0}\lim_{\mathcal{L}\to\infty} \frac{1}{2\beta}\frac{1}{\mathcal{L}}\sum_{n=1}^{\mathcal{L}} \bar{\eta}(n)\Big[J\big(\theta(n) + \beta\bar{\eta}(n)\big) - J\big(\theta(n) - \beta\bar{\eta}(n)\big)\Big] \tag{18}$$

where $\bar{\eta}(n) = (\bar{\eta}_1(n), \ldots, \bar{\eta}_K(n))^T, n \geq 0$ are assumed to be vectors of independent $N(0, 1)$ random variables. Equation (18) can be approximated for a large enough $\mathcal{L}$ and small $\beta > 0$ to:

$$\nabla J(\theta(n)) \approx \frac{1}{2\beta}\frac{1}{\mathcal{L}}\sum_{n=1}^{\mathcal{L}} \bar{\eta}(n)[J\big(\theta(n) + \beta\bar{\eta}(n)\big) - J\big(\theta(n) - \beta\bar{\eta}(n)\big)] \tag{19}$$

It has been shown in [21], [33] and [18] that two-simulation gradient estimator perform better than one-simulation. We are using a different two-simulation estimator that was used in [22] which is:

$$\nabla J(\theta(n)) \approx \frac{1}{\beta}\frac{1}{\mathcal{L}}\sum_{n=1}^{\mathcal{L}} \bar{\eta}(n)\left[J\left(\theta(n) + \beta\bar{\eta}(n)\right) - J\left(\theta(n)\right)\right] \tag{20}$$

where the convolution is given by:

$$D_{\beta,2}J(\theta) = \mathbb{E}\left[\frac{\bar{\eta}}{2\beta}\left(J(\theta + \beta\bar{\eta}) - J(\theta - \beta\bar{\eta})\right)\right] \tag{21}$$

Note that, 2 in subscript of operator D indicates that it corresponds to two-simulation estimation.

B. NESTEROV'S ACCELERATION - MOMENTUM BASED METHODS

One of the most popular first-order iterative optimization algorithms is Stochastic Gradient Descent. The update rule for the algorithm is given by:

$$\theta(n+1) = \theta(n) - a(n)\nabla J(\theta(n)) \tag{22}$$

where $a(n)$ is the step size.

As can be inferred from equation (22), the update is directly dependent on the gradient. Hence, when the slope is too flat or noisy, it can take a long time to converge. The momentum-based gradient descent method is used to overcome these problems. This algorithm was first introduced by Polyak [11]. In the momentum-based technique, when the gradients of consecutive iterations are in the same direction, the algorithm takes giant steps. Hence, it gains leverage of the gradients of previous iterations in the update rule to accelerate gradient descent. Polyak's momentum update is given by:

$$\theta(n+1) = \theta(n) - a(n)\nabla J(\theta(n)) + \zeta(\theta(n) - \theta(n-1)) \tag{23}$$

where ζ is a hyperparameter, which scales down the previous step. However, this method struggles when the function to optimize is highly convex [34] as the update overshoots desired minimum again and again. Nesterov's accelerated gradient descent [13] algorithm solves this problem by modifying the update in following way:

$$\theta(n+1) = \theta(n) + \zeta(\theta(n) - \theta(n-1)) - a(n)\nabla J(\theta(n) + \zeta(\theta(n) - \theta(n-1))) \tag{24}$$

It is clear from the Nesterov's update rule that momentum is applied before gradient evaluation unlike Polyak's momentum method. This look ahead move in the gradient of Nesterov's update avoids overshoots by reducing the momentum when overshoot happens. We use the version of Nesterov's update as mentioned in [35]:

$$\theta(n) = y(n-1) - a(n-1)\nabla J(y(n-1)),$$
$$y(n) = \theta(n) + \frac{n-1}{n+2}\Big(\theta(n) - \theta(n-1)\Big).$$

where $\theta(0)$ is chosen randomly and $y(0) = \theta(0)$. Consider $n = n+1$ in the above equation:

$$\theta(n+1) = y(n) - a(n)\nabla J(y(n)), \tag{25}$$

$$y(n+1) = \theta(n+1) + \frac{n}{n+3}\Big(\theta(n+1) - \theta(n)\Big). \tag{26}$$

IV. SF WITH NESTEROV'S ACCELERATION

In this section we give the SF algorithm with Nesterov's acceleration. Let $\eta = (\eta_1, \ldots, \eta_K)^\top$ with each element, say η_j be an independent random variable taking values in ± 1 with probability $1/2$. Let $\beta > 0$ be a small constant.

Algorithm 1 NAG-SF Algorithm

```
Initialize $Z_l(0) = 0$, $\theta_l(0) = 0$, $l = 1, \ldots, K$.
Fix (large) integer $\mathcal{L}$ and set $n = 0$.
while $n < \mathcal{L}$ do
    Generate $X_n$ and $X'_n$ independently as different simulation samples from parameters $y(n)$ and $y(n) + \beta\eta(n)$.
    Then $\forall\, l = 1, \ldots, K$, update
        $Z_l(n+1) = Z_l(n) + b(n)\Big(\frac{\eta_l(n)}{\beta}\Big(h\big(X'_n\big) - h\big(X_n\big)\Big) - Z_l(n)\Big)$,   (27)
        $\theta_l(n+1) = \Gamma\left(y_l(n) - a(n)Z(n)\right)$,   (28)
        $y_l(n+1) = \theta_l(n+1) + \frac{n}{n+3}\Big(\theta_l(n+1) - \theta_l(n)\Big)$.   (29)
    Set $n := n + 1$
end while
Output $\theta(n) = (\theta_1(n), \ldots, \theta_d(n))^T$ and terminate.
```

There are two recursions defined in the algorithm. One is in equation (27) and other in equation (29)) which are driven by step size parameter $a(n)$ and $b(n)$ respectively. The assumption of preventing premature convergence of step sizes is standard in stochastic approximation algorithms, along with the assumption of asymptotic decrease as given in equation (8).

Let $\theta = (\theta_1, \ldots, \theta_K)^T$ denote the parameter vector. $\Gamma = (\Gamma_1, \ldots, \Gamma_K)^T$ is mapping that projects θ onto the compact and convex set C i.e. $\Gamma : \mathbb{R}^K \to C \subset \mathbb{R}^K$. This type of projection is generally considered to be a non-trivial task, however, in certain problems, such as queue routing problem as in Section VI, the projection set C is taken to be a hyper rectangle of the form $C = \prod_{i=1}^{K}[L_{i,\min}, L_{i,\max}]$. Here, the interval $[L_{i,\min}, L_{i,\max}]$ is the projection space to which θ_i (the ith component of θ) is projected. In other words $\Gamma_i(\theta_i) = \min(L_{i,\max}, \max(\theta_i, L_{i,\min}))$.

V. CONVERGENCE ANALYSIS

Let a sequence of σ-fields be defined as $\mathcal{F}(k) = \sigma(\theta_i(n), X_n, X'_n, n \leq k, \eta_i(n), n < k, i = 1, \ldots, K), k \geq 1$. For a fixed $\beta > 0$ the estimated gradient (cf. (27)) in NAG-SF algorithm is assumed to be defined as $Z(n) = (Z_l(n), \forall l = 1, \ldots, K)^T$. Next, $Q_l(n)$ is defined as:

$$Q_l(n) = \sum_{m=1}^{n} b(m)\Bigg(\frac{\eta_l(m)}{\beta}\left(h(X'_m) - h(X_m)\right) - \mathbb{E}\Big(\frac{\eta_l(m)}{\beta}(h(X'_m) - h(X_m))|\mathcal{F}(m-1)\Big)\Bigg), \qquad (30)$$

where $l = 1, \ldots, K, n \geq 1$.

Lemma 2: Sequences $\{Q_l(n), \mathcal{F}(n)\}, l = 1, \ldots, K$ are almost surely convergent martingales.

Consider the following system of ordinary differential equations (ODEs):

$$\dot{\theta}(t) = 0, \qquad (31)$$

$$\dot{Z}(t) = D_{\beta,2}J(\theta(t)) - Z(t), \qquad (32)$$

where $D_{\beta,2}$ operator is defined in equation (19).

For $\tau > 0, \mu > 0$, we call $y(\cdot)$ a (τ, μ)-perturbation of the o.d.e. $\dot{x}(t) = \mathcal{F}(x(t))$ (with G as an asymptotically stable attracting set). If there exists an increasing sequence $\{\tau_i, i \geq 0\}$ of real numbers with $\tau_0 = 0$ and $\forall\, i$, $\tau_{i+1} - \tau_i \geq \tau$, such that, on each interval $[\tau_i, \tau_{i+1}]$, there exists a solution $x^i(\cdot)$ of the above o.d.e. such that

$$\sup_{t \in [\tau_i, \tau_{i+1}]} |x^i(t) - y(t)| < \mu.$$

Let G^ϵ denote the ϵ-neighbourhood of a set G, i.e., $G^\epsilon = \{x | \exists\, x' \in G \text{ such that } ||x - x'|| < \epsilon\}$. We now recall a result from Hirsch [36] stated as the next Lemma. (Theorem 1, pp. 339).

Lemma 3: Given $\epsilon > 0, \tau > 0$, there exists a $\bar{\mu} > 0$ such that, for all $\mu \in [0, \bar{\mu}]$, any (τ, μ) -perturbation of $\dot{x}(t) = \mathcal{F}(x(t))$ converges to G^ϵ.

Lemma 4: The sequence of updates $\{Z(p)\}$ is uniformly bounded with probability one.

The proof for Lemma 2-4, we refer the readers to the Appendix of [22]. These lemmas are required to show that the noise term is bounded.

Assume that $r(n) = \sum_{i=0}^{n-1} b(i), n \geq 1$. Consider the function $\hat{Z}(t)$ defined according to $\hat{Z}(r(n)) = Z(n)$ with maps $t \to \hat{Z}(t)$ corresponding to continuous linear interpolations on the intervals $[r(n), r(n+1)]$. Given $T > 0$, define $\{T_n\}$ as follows: $T_0 = 0$ and for $n \geq 1$, $T_n = \min r(m) | r(m) \geq T_{n-1} + T$. Let $I_n = [T_n, T_{n+1})$. Note that, there exists some integer $m_n > 0$ such that $T_n = r(m_n)$. Define also functions $Z^n(t), t \in I_n, n \geq 0$, that are obtained as trajectories of the o.d.e.

$$\dot{Z}^n(t) = D_{\beta,2}J(\theta) - Z^n(t), \qquad (33)$$

with $Z^n(T_n) = \hat{Z}(r(m_n)) = Z(m_n)$. Now, note that one can rewrite equation (28) as follows:

$$\theta(n+1) = \Gamma(\theta(n) + b(n)\varepsilon_1(n)) \qquad (34)$$

where $\varepsilon_1(n) = -\frac{a(n)}{b(n)}M(n)Z(n) \to 0$ as $n \to \infty$ almost surely. Let $\theta(t)$ be defined as: $\theta(r(n)) = \theta(n), n \geq 0$, and $\theta(t)$, for $t \in [r(n), r(n+1)]$ is a continuous linear interpolation between $\theta(n)$ and $\theta(n+1)$. Now, for $\gamma > 0$, $\theta(r(n) + \cdot)$ can be seen to be a bounded (T, γ)-perturbation of the o.d.e. $\dot{\theta}(t) = 0$ for a sufficiently large n. In other words, θ can be assumed to be fixed (i.e., $\theta(t) = \theta\ \forall t$) when viewed from the time scale of $\{b(n)\}$ or that the parameter update recursion is quasi-static.

Using a standard argument based on Gronwall's inequality, it can now be shown that:

Lemma 5:

$$\lim_{n\to\infty} \sup_{t\in I_n} ||Z^n(t) - \hat{Z}(t)|| = 0 \text{ w.p. } 1.$$

Proof: The proof requires the results from lemma 2 and 4. For details we refer the readers to lemma 1, chapter 2 of [23]. □

Next, we have the following result.

Lemma 6: Given $T, \gamma > 0$, $\big((\theta(r(n)+\cdot), Z(r(n)+\cdot)\big)$, is a bounded (T, γ)-perturbation of equation (31) *and equation* (32) *for a sufficiently large n.*

Proof: Since the parameter recursion can be written as in equation (34), the claim follows from Lemma 5. □

Lemma 7:

$$||Z(n) - D_{\beta,2}J(\theta(n))|| \to 0 \text{ w.p. } 1 \text{ as } n \to \infty.$$

Proof: The claim follows by applying Lemma 3 on o.d.e. (32) for every $\epsilon > 0$. □

The following result shows that the gradient estimates are unbiased in the limit as $\beta \to 0$.

Lemma 8: $\lim_{\beta\to 0} \lim_{n\to\infty} ||D_{\beta,2}J(\theta(n)) - \nabla J(\theta(n))|| = 0$ *w.p.* 1.

Proof: Refer Proposition A.14 of [18]. □

Proposition 1: $\lim_{\beta\to 0} \lim_{n\to\infty} ||Z(n) - \nabla J(\theta(n))|| = 0$ *w.p.* 1.

Proof: The claim follows from Lemmas 7 and 8 using the triangle inequality. □

Next, consider the slower time scale recursion. Define $t(n) = \sum_{i=0}^{n-1} a(i), n \geq 1$. Consider the function $\hat{M}(t)$ defined according to $\hat{M}(t(n)) = M(n)$ with maps $t \to \hat{M}(t)$ corresponding to continuous linear interpolations on intervals $[t(n), t(n+1))$. Now consider the o.d.e.

$$\dot{\theta}(t) = \tilde{\Gamma}(-\hat{M}(t)\nabla J(\theta(t))), \tag{35}$$

where for any $y \in \mathbb{R}^N$ and a bounded, continuous function $v(\cdot) : \mathbb{R}^N \to \mathbb{R}^N$,

$$\tilde{\Gamma}(v(y)) = \lim_{\eta\to 0} \frac{(\Gamma(y + \eta v(y)) - \Gamma(y))}{\eta}. \tag{36}$$

Also consider that for $y \in C^0$, where C^0 denote the interior of C, $\tilde{\Gamma}(v(y)) = v(y)$. Also, for $y \in \partial C$, the boundary of C, such that $y+\eta v(y) \notin C$ for any $\eta > 0$, $\tilde{\Gamma}(v(y))$ is the projection of $v(y)$ to C. Note also that the limit in equation (36) is well defined because C is assumed to be a compact and convex set. In case the limit is not well defined, one may replace it with the set of all limit points there. The corresponding o.d.e. in (35) will then become a differential inclusion.

Assumption 11: The Markov chain $\{X_n\}$ under any stationary randomized policy π is irreducible.

Assumption 12: The basis functions $\{f(k), k = 1, \ldots, d_1\}$ are linearly independent. Further, $d_1 \leq |S|$ and Φ has full rank.

Theorem 4: Let Assumptions 11-12 hold. Then given $\epsilon > 0$, $\exists\, \beta_0 > 0$ such that for all $\beta \in (0, \beta_0)$, $\theta(n), n > 0$ obtained according to the equations (28), (29) *satisfy $\theta(n) \to K^\epsilon$ as $n \to \infty$ with probability one.*

Proof: Along the slower time scale of $b(n)$, we can rewrite the θ-recursion as

$$\theta_l(n+1) = \Gamma\left(y_l(n) - b(n)\left(\frac{a(n)}{b(n)}\nabla J(\theta) + \varepsilon_1(n)\right)\right)$$

where $\varepsilon_1(n) \to 0$ as $n \to \infty$.

Next we have for $\theta \in C^0$ the o.d.e. for Nesterov's scheme i.e., equations (28) and (29) (see equations (1) and (3) from [35]) as

$$\ddot{\theta} + \frac{3}{t}\dot{\theta} + D_{\beta,2}J(\theta) = 0. \tag{37}$$

Consider

$$\ddot{\theta} + \frac{3}{t}\dot{\theta} + \nabla J(\theta) = 0. \tag{38}$$

As, $\beta \to 0$, trajectories of the o.d.e. in equation (38) converge to those of equation (37) uniformly on compacts when starting in the same initial conditions for both (see proof of Theorem 2 in [37]).

Now, equation (38) can be converted to a first order o.d.e. by taking $\dot{\theta} = \varsigma$. We have $\dot{\varsigma} + \frac{3}{t}\varsigma + \nabla J(\theta) = 0$. with $\mathcal{X} = \begin{bmatrix}\theta \\ \varsigma\end{bmatrix}$, we have

$$\dot{\mathcal{X}} = \begin{bmatrix}\dot{\theta} \\ \dot{\varsigma}\end{bmatrix} = \begin{bmatrix}\varsigma \\ -\frac{3}{t}\varsigma - \nabla J(\theta)\end{bmatrix}.$$

We start the system at $t = t_0$. Define a Lyapunov function $Z(\cdot)$ according to

$$Z^t(\mathcal{X}) = \left(\frac{1}{2}\varsigma^2 + J(\theta)\right)\left(1 + \frac{1}{t}\right)$$

It can be seen that $Z^t(\mathcal{X}) > 0$. Then corresponding to the o.d.e. (38), we have

$$\begin{aligned}\frac{dZ^t(\mathcal{X})}{dt} &= (\varsigma \cdot \dot{\varsigma} + \nabla J(\theta)\dot{\theta})\left(1 + \frac{1}{t}\right) + \left(\frac{1}{2}\varsigma^2 + J(\theta)\right)\frac{-1}{t^2} \\ &= \left(\varsigma \cdot \left(-\frac{3}{t}\varsigma - \nabla J(\theta)\right) + \varsigma\nabla J(\theta)\right)\left(1 + \frac{1}{t}\right) \\ &\qquad - \left(\frac{1}{2}\varsigma^2 + J(\theta)\right)\frac{1}{t^2} \\ &= \frac{-3}{t}\varsigma^2\left(1 + \frac{1}{t}\right) - \left(\frac{1}{2}\varsigma^2 + J(\theta)\right)\frac{1}{t^2} < 0\end{aligned} \tag{39}$$

As $J(\theta)$ was assumed to be in $\mathbb{R}^+$. □

Consider a non-autonomous system

$$\dot{x} = Q(t, x) \tag{40}$$

where $Q : [0, \infty) \times D \to \mathbb{R}^d$ is piece-wise continuous in t and locally Lipschitz in x on $[0, \infty) \times D$ and $D \subset \mathbb{R}^d$. For this system, the equilibrium point $x = 0$ is uniformly asymptotically stable if:

- for each $\varepsilon > 0$, $\exists$ a constant δ dependent on only ε and $\delta(\varepsilon) > 0$ such that

$$||x(t_0)|| < \delta \Rightarrow ||x(t)|| < \varepsilon, \quad \forall t \geq t_0 \geq 0 \tag{41}$$

- there exist a scalar $c > 0$, independent of t_0, such that $\forall ||x(t_0)|| < c, x(t) \to 0$ as $t \to \infty$, uniformly in t_0; that is, for each $\alpha > 0$ there is $T = T_\alpha > 0$ such that

$$||x(t)|| < \alpha, \quad \forall t \geq t_0 + T_\alpha, \ \forall ||x(t_0)|| < c \tag{42}$$

Theorem 5 (Theorem 4.9 of [38]): Let $x = 0$ be an equilibrium point for $\dot{x} = f(t, x)$ and $D \subset \mathbb{R}^n$ be a domain containing $x = 0$. Let $W_1(x)$, $W_2(x)$ and $W_3(x)$ be continuous positive definite functions on D. Also let $V : [0, \infty) \to \mathbb{R}$ be a continuously differentiable function such that

$$W_1(x) \leq V(t, x) \leq W_2(x) \tag{43}$$

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t, x) \leq -W_3(x) \tag{44}$$

for all $t \geq 0$ and $\forall x > D$. Then $x = 0$ is uniformly asymptotically stable.

Proof:

In our case the Lyapunov function is $V(t, x) = Z^t(\mathcal{X})$. The first condition in equation (43) is satisfied with $W_2(\mathcal{X}) = (\frac{1}{2}\varsigma^2 + \nu^{\theta^\top} V)(1 + 1/t_0)$ and $W_1(\mathcal{X}) = (\frac{1}{2}\varsigma^2 + \nu^{\theta^\top} V)$. The second condition in equation (44) can be seen from equation (39). Thus the system is uniformly asymptotically stable. The claim follows from Theorem 1, pp. 339 of [36]. The proof of this theorem uses the boundedness of the system (see the $\Gamma(\cdot)$ operator from equation (28)) to show that the perturbed trajectory is also bounded. Though this theorem in [36] is for autonomous o.d.e., the same proof goes through for non-autonomous o.d.e. also.

The asymptotically stable equilibria of o.d.e. $\dot{\mathcal{X}} = 0$ is the set where $\varsigma = 0$ and $\frac{3}{t}\varsigma + D_{\beta,2}J(\theta) = 0$ which corresponds to the set K within the set C. These can be seen to correspond to the local minima of the function $D_{\beta,2}J(\theta)$. □

This establishes the convergence of the NAG-SF algorithm.

VI. EXPERIMENTAL RESULTS AND DISCUSSION

To test the performance of NAG-SF, we compare it with three different algorithms, namely: Quasi-Newton SF (QN-SF) [22], Gradient SF (G-SF) and Jacobi Variant of Newton SF (JN-SF) [18] on three different problems.

A. QUADRATIC LOSS MINIMIZATION

First is the minimization of a very simple quadratic loss function $L(\theta)$ considered by Zhu and Spall [39] which is given by

$$L(\theta) = \frac{1}{2}\theta^T H \theta \tag{45}$$

We have used the stochastic version of this function: $y(\theta) = L(\theta) + Y$ where Y is Gaussian Noise $N(0, \sigma^2)$. We set $c_1 = 0.1291$ and $c_2 = 1.1311$ to define Hessian H which is a 4×4 matrix as:

$$H_{ij} = c_1 \exp[-(i - j)^2 / c_2^2] \tag{46}$$

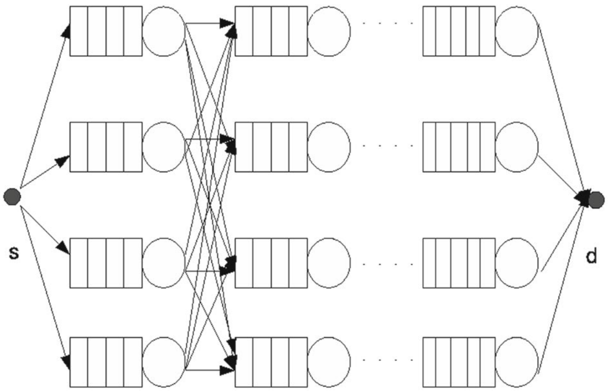

FIGURE 1. Multi-stage queuing network.

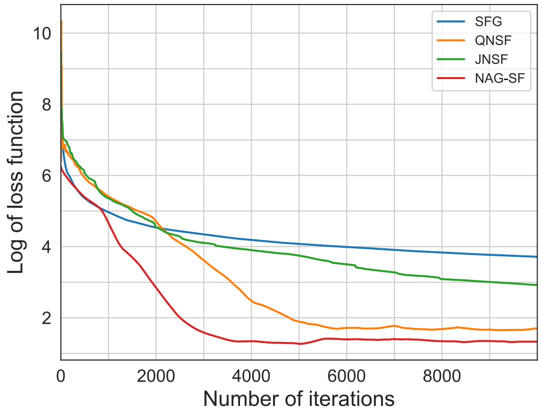

FIGURE 2. Simulation results for quadratic loss function.

At $\theta^* = 0$, $L(\theta^*) = 0$, which is the minimum for this function.

B. ROUTING PROBLEM IN QUEUING NETWORKS

To demonstrate the efficiency of the proposed setup in a real-world example, we show the problem of finding optimal routing probabilities in a multi-stage queuing network. This network is considered to replicate a simplified model of the internet [22], [24], as shown in figure 1.

The network consists of Z servers, one source s, and one sink d. Here, packets arriving at the source s are immediately routed to reach the sink d through intermediate servers. When a server is serving a packet, then that arrived packet is pushed in a queue if another packet arrives. Packet waits till the server becomes free to execute it. There is an exponential queuing delay (service time) at each node (server), and the arrival time is Poisson distributed. Therefore, the queue length forms a Markov Process. It is assumed that there is no delay in transmission; thus, a packet immediately arrives at the next node after being routed by the previous node. The routing probability vector for nth update is denoted by parameter $\theta(n)$. The objective is to minimize the long-run average end-to-end delay of each packet $J(\theta)$. Here, the cost function h denotes the total end-to-end delay. These Z servers are arranged in stages, where each stage has N servers, and there are M stages between source and sink. Each server in stage i

TABLE 1. Expected delay over 500 iterations for various combinations of λ and μ with single stage.

λ/μ	10/5	5/10	5/5	10/10	15/15
Optimum	0.91 ± 0.000	0.25 ± 0.000	0.49 ± 0.000	0.43 ± 0.000	0.41 ± 0.000
G-SF	1.13 ± 0.041	0.33 ± 0.081	0.63 ± 0.052	0.61 ± 0.063	0.57 ± 0.082
QN-SF	0.99 ± 0.089	0.50 ± 0.072	0.70 ± 0.075	0.56 ± 0.076	0.48 ± 0.079
JN-SF	1.06 ± 0.066	0.42 ± 0.075	0.62 ± 0.068	0.53 ± 0.083	0.55 ± 0.081
NAG-SF	0.97 ± 0.037	0.30 ± 0.068	0.59 ± 0.052	0.51 ± 0.038	0.46 ± 0.062

TABLE 2. Expected delay over 500 iterations for various combinations of λ and μ with three stages.

λ/μ	10/5	5/10	5/5	10/10	15/15
Optimum	1.67 ± 0.000	0.56 ± 0.000	1.20 ± 0.000	0.79 ± 0.000	0.65 ± 0.000
G-SF	3.12 ± 0.045	1.62 ± 0.053	2.55 ± 0.066	1.48 ± 0.074	1.83 ± 0.082
QN-SF	3.88 ± 0.036	1.19 ± 0.075	3.76 ± 0.071	1.96 ± 0.066	1.76 ± 0.057
JN-SF	3.49 ± 0.079	1.12 ± 0.098	3.82 ± 0.084	2.14 ± 0.089	1.59 ± 0.072
NAG-SF	3.01 ± 0.035	1.08 ± 0.071	2.05 ± 0.056	1.42 ± 0.062	1.23 ± 0.063

TABLE 3. Expected delay over 500 iterations for various combinations of λ and μ with five stages.

λ/μ	10/5	5/10	5/5	10/10	15/15
Optimum	2.24 ± 0.000	0.98 ± 0.000	2.11 ± 0.000	1.33 ± 0.000	0.92 ± 0
G-SF	3.54 ± 0.063	2.01 ± 0.063	3.14 ± 0.078	1.96 ± 0.064	2.09 ± 0.065
QN-SF	4.29 ± 0.057	1.58 ± 0.079	3.72 ± 0.064	2.44 ± 0.075	2.02 ± 0.066
JN-SF	3.91 ± 0.086	1.51 ± 0.086	3.78 ± 0.092	2.62 ± 0.088	1.85 ± 0.032
NAG-SF	3.43 ± 0.051	1.47 ± 0.061	2.94 ± 0.055	1.85 ± 0.073	1.39 ± 0.054

TABLE 4. Expected delay over 500 iterations for various combinations of λ and μ with seven stages.

λ/μ	10/5	5/10	5/5	10/10	15/15
Optimum	2.78 ± 0.000	1.34 ± 0.000	2.61 ± 0.000	1.74 ± 0.000	1.17 ± 0.000
G-SF	4.08 ± 0.057	2.03 ± 0.078	3.70 ± 0.032	2.16 ± 0.081	2.29 ± 0.060
QN-SF	4.85 ± 0.081	1.91 ± 0.034	4.21 ± 0.031	2.64 ± 0.095	2.22 ± 0.085
JN-SF	4.45 ± 0.052	1.84 ± 0.076	4.17 ± 0.043	2.82 ± 0.070	2.05 ± 0.098
NAG-SF	3.98 ± 0.046	1.79 ± 0.041	3.20 ± 0.047	2.10 ± 0.046	1.69 ± 0.061

is connected to all the servers in the previous stage $i-1$ and next stage $i+1$ (except for servers of first and last stage as they are connected to source s and sink d respectively). The total number of servers is fixed in each stage.

There is no delay from the source to servers of the first stage. A packet is routed from one stage to another following routing probabilities. The algorithm calculates optimal routing probabilities $\theta(n)$ for minimizing queuing delay. Our goal is to optimize $J(\theta)$ which is the long-run average cost, by obtaining $\theta^* \in C$ s.t. $\theta^* = \arg\min J(\theta) | \theta \in C$.

C. CLIMATE PREDICTION

We use the SF algorithms as the optimization algorithm along with a sigmoid function to create a modified version of logistic regression, and then apply it to a climate prediction problem. The flowchart for the classification algorithm is in figure 3. We use l2-regularized logistic loss as loss function $h(X)$. The predicted value $\hat{y}$ is calculated by using a sigmoid function and is given by:

$$\hat{y} = \frac{1}{1+\exp^{-(\theta_0+\theta_1\, s_1+\ldots+\theta_K s_k)}}$$

for which the loss function $h(\theta)$ is defined as:

$$h(\theta) = \sum -y\log(\hat{y}) - (1-y)\log(1-\hat{y}) + r_e \sum(\theta)$$

where y denotes the true value of the label which is either 0 or 1 and the regularization rate r_e is set to 5. The dataset is taken from NCAR [27] for one year $(2008-2009)$ or 365 days collected from 2500 stations of which 1250 stations corresponds to tropical climate and rest to polar climate. The total data samples are $912,500$. The train test split is done in a ratio of 80:20. The training data is further split in a ratio of 80:20 for validation. We use climate classification by Koppen-Geiger [40] to classify the dataset into two classes:

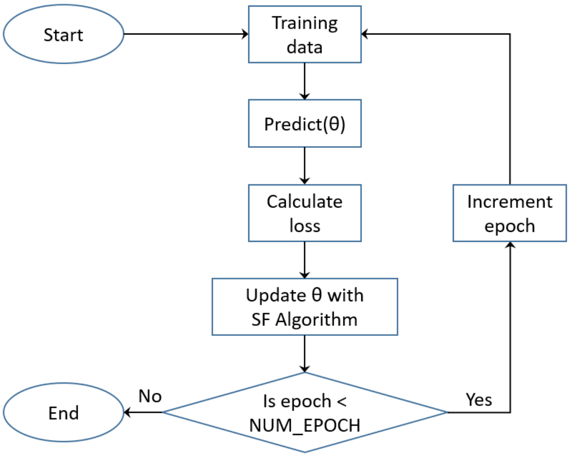

FIGURE 3. Flowchart of SF based classification algorithm.

TABLE 5. Performance of SF algorithms on weather data.

	Accuracy	F-Score	Precision	Recall
G-SF	69.94	0.7273	0.7432	0.7121
QN-SF	76.40	0.7895	0.7962	0.7830
JN-SF	71.85	0.7590	0.7479	0.7704
NAG-SF	79.32	0.8004	0.7926	0.8084

tropical climate and polar climate. Thus it is a two-class classification problem on real data with the following features considered against each record: average humidity, average temperature and precipitation.

D. RESULTS

For all the experiments NAG-SF outperformed SF without acceleration. In the quadratic loss function minimization, parameters used are $\mathcal{L} = 100000$, $\beta = 0.9$, $a(n) = (n+1)^{-1}$, $b(n) = (n+1)^{-0.10}$. In figure 2, it can be seen that the NAG-SF approaches desired solution faster than other SF algorithms. In routing problem, the parameters for the algorithm are $\mathcal{L} = 500$, $\beta = 0.9$, $a(n) = (n+1)^{-1}$, $b(n) = (n+1)^{-0.95}$. We have calculated the performance of algorithms by varying the number of stages to one, three, five and seven with 4 nodes at each stage and the results are shown in Table 1-4. We also added another step of normalization of θ after the update to bound the routing probability matrix. Similar performance was observed in classification problem on weather dataset. The parameters for NAG-SF are $\beta = 0.95$ and $a(n) = (n+1)^{-1}$, $b(n) = (n+1)^{-0.85}$, but the step-sizes are changed per 100 steps and the number of epochs are 300. The experimental results are shown in Table 5 which shows that NAG-SF performed better than other baselines.

VII. CONCLUSION

In this paper, we proposed a smoothed functional algorithm with Nesterov's acceleration for unconstrained minimization problems. We then presented the proof for convergence of the algorithm and experimental results for (a) quadratic loss function minimization, (b) optimal routing in a multi-stage queuing network problem and (c) climate prediction. Numerical results verified that our proposed framework performed better than other smoothed functional algorithms.

One possible future work for the analysis of NAG-SF could be to provide a result for the convergence rate of this algorithm, which we may take as our next endeavour.

REFERENCES

[1] E. K. Chong and S. H. Zak, *An Introduction to Optimization*. Hoboken, NJ, USA: Wiley, 2004.

[2] J. Schneider and S. Kirkpatrick, *Stochastic Optimization*. Berlin, Germany: Springer-Verlag, 2007.

[3] G. Lan, *First-Order and Stochastic Optimization Methods for Machine Learning*. Springer, 2020.

[4] X. Ye, D. Ge, X. Bian, Q. Xu, and Y. Zhou, "Improving business process efficiency for supply chain finance: Empirical analysis and optimization based on stochastic Petri net," *IEEE Access*, vol. 8, pp. 98430–98448, 2020.

[5] J. Peng, B. Zhang, and S. Li, "Towards uncertain network optimization," *J. Uncertainty Anal. Appl.*, vol. 3, no. 1, pp. 1–19, Dec. 2015.

[6] B. Zhang, H. Li, S. Li, and J. Peng, "Sustainable multi-depot emergency facilities location-routing problem with uncertain information," *Appl. Math. Comput.*, vol. 333, pp. 506–520, Sep. 2018.

[7] B. Zhang, J. Peng, and S. Li, "Covering location problem of emergency service facilities in an uncertain environment," *Appl. Math. Model.*, vol. 51, pp. 429–447, Nov. 2017.

[8] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*. [Online]. Available: http://arxiv.org/abs/1609.04747

[9] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, pp. 400–407, Sep. 1951.

[10] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 315–323.

[11] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Comput. Math. Math. Phys.*, vol. 4, no. 5, pp. 1–17, 1964.

[12] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1139–1147.

[13] Y. E. Nesterov, "A method for solving the convex programming problem with convergence rate $O(1/k^2)$," *Doklady Akademii Nauk SSSR*, vol. 25, pp. 543–547, 1983. [Online]. Available: https://www.esaim-cocv.org/articles/cocv/abs/2019/01/cocv170092/cocv170092.html

[14] H. Attouch, Z. Chbani, and H. Riahi, "Rate of convergence of the Nesterov accelerated gradient method in the subcritical case $\alpha \leq 3$," *ESAIM, Control, Optim. Calculus Variat.*, vol. 25, p. 2, Jun. 2019.

[15] R.-E. Plessix, "A review of the adjoint-state method for computing the gradient of a functional with geophysical applications," *Geophys. J. Int.*, vol. 167, no. 2, pp. 495–503, Nov. 2006.

[16] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function," *Ann. Math. Stat.*, vol. 23, no. 3, pp. 462–466, 1952.

[17] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Trans. Autom. Control*, vol. 37, no. 3, pp. 332–341, Mar. 1992.

[18] S. Bhatnagar, "Adaptive Newton-based multivariate smoothed functional algorithms for simulation optimization," *ACM Trans. Model. Comput. Simul.*, vol. 18, no. 1, p. 2, 2007.

[19] S. Bhatnagar, H. Prasad, and L. Prashanth, *Stochastic Recursive Algorithms for Optimization: Simultaneous Perturbation Methods*, vol. 434. London, U.K.: Springer-Verlag, 2012.

[20] V. Y. Katkovnik and O. Y. Kulchits, "Convergence of a class of random search algorithms," *Autom. Remote Control*, vol. 33, no. 8, pp. 1321–1326, 1972.

[21] M. A. Styblinski and T.-S. Tang, "Experiments in nonconvex optimization: Stochastic approximation with function smoothing and simulated annealing," *Neural Netw.*, vol. 3, no. 4, pp. 467–483, Jan. 1990.

[22] K. Lakshmanan and S. Bhatnagar, "Quasi-Newton smoothed functional algorithms for unconstrained and constrained simulation optimization," *Comput. Optim. Appl.*, vol. 66, no. 3, pp. 533–556, Apr. 2017.

[23] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*, vol. 48. New Delhi, India: Hindustan Book Agency, 2009.

[24] K. Lakshmanan and B. Shalabh, ''Smoothed functional and quasi-Newton algorithms for routing in multi-stage queueing network with constraints,'' in *Proc. Int. Conf. Distrib. Comput. Internet Technol.* Berlin, Germany: Springer, 2011, pp. 175–186.

[25] D. Karaboga and B. Basturk, ''On the performance of artificial bee colony (ABC) algorithm,'' *Appl. Soft Comput.*, vol. 8, no. 1, pp. 687–697, 2008.

[26] S. Gao and I. Chabini, ''Optimal routing policy problems in stochastic time-dependent networks,'' *Transp. Res. B, Methodol.*, vol. 40, no. 2, pp. 93–122, 2006.

[27] Climate Prediction Center, National Centers for Environmental Prediction, National Weather Service, NOAA, U.S. Department of Commerce, ''CPC global summary of day/month observations,'' Res. Data Arch. Nat. Center Atmos. Res., Comput. Inf. Syst. Lab., Boulder, CO, USA, 1987. [Online]. Available: https://rda.ucar.edu/datasets/ds512.0/

[28] S. Bhatnagar, H. Prasad, and L. Prashanth, ''Stochastic approximation algorithms,'' in *Stochastic Recursive Algorithms for Optimization*. London, U.K.: Springer, 2013, pp. 17–28.

[29] M. Benaim, ''A dynamical system approach to stochastic approximations,'' *SIAM J. Control Optim.*, vol. 34, no. 2, pp. 437–472, Mar. 1996.

[30] V. S. Borkar and S. P. Meyn, ''The ODE method for convergence of stochastic approximation and reinforcement learning,'' *SIAM J. Control Optim.*, vol. 38, no. 2, pp. 447–469, 2000.

[31] V. S. Borkar and V. R. Konda, ''The actor-critic algorithm as multi-time-scale stochastic approximation,'' *Sadhana*, vol. 22, no. 4, pp. 525–543, Aug. 1997.

[32] S. Bhatnagar and V. S. Borkar, ''Multiscale chaotic SPSA and smoothed functional algorithms for simulation optimization,'' *Simulation*, vol. 79, no. 10, pp. 568–580, Oct. 2003.

[33] D. C. Chin, ''Comparative study of stochastic algorithms for system optimization based on gradient approximations,'' *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 27, no. 2, pp. 244–249, Apr. 1997.

[34] L. Lessard, B. Recht, and A. Packard, ''Analysis and design of optimization algorithms via integral quadratic constraints,'' *SIAM J. Optim.*, vol. 26, no. 1, pp. 57–95, Jan. 2016.

[35] W. Su, S. Boyd, and E. Candes, ''A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2510–2518.

[36] M. W. Hirsch, ''Convergent activation dynamics in continuous time networks,'' *Neural Netw.*, vol. 2, no. 5, pp. 331–349, 1989.

[37] S. Bhatnagar, ''An actor–critic algorithm with function approximation for discounted cost constrained Markov decision processes,'' *Syst. Control Lett.*, vol. 59, no. 12, pp. 760–766, Dec. 2010.

[38] H. K. Khalil, *Nonlinear Systems*. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[39] X. Zhu and J. C. Spall, ''A modified second-order SPSA optimization algorithm for finite samples,'' *Int. J. Adapt. Control Signal Process.*, vol. 16, no. 5, pp. 397–409, 2002.

[40] M. Kottek, J. Grieser, C. Beck, B. Rudolf, and F. Rubel, ''World map of the Köppen-Geiger climate classification updated,'' *Meteorologische Zeitschrift*, vol. 15, pp. 259–263, May 2006, doi: 10.1127/0941-2948/2006/0130.

ABHINAV SHARMA received the M.Tech. degree in system science engineering from Indian Institute of Technology (IIT) at Jodhpur, Jodhpur, India, in 2015. He is currently pursuing the Ph.D. degree in stochastic optimization with Indian Institute of Information Technology at Jabalpur, Jabalpur, India.

K. LAKSHMANAN received the Ph.D. degree in computer science and automation from Indian Institute of Science Bangalore, India, in 2013. He has experience working as a Postdoctoral Fellow at IIT Bombay, India, and the National University of Singapore, Singapore. He is currently an Assistant Professor of computer science and engineering at Indian Institute of Technology (BHU) at Varanasi, Varanasi, India. His research interests include machine learning, stochastic optimization, and reinforcement learning.

RUCHIR GUPTA (Senior Member, IEEE) received the Ph.D. degree in peer-to-peer networks from IIT Kanpur, India, in 2013. From 2017 to 2020, he served as an Associate Professor with IIT Banaras Hindu University (BHU), Varanasi, India. He is currently working as a Professor with the Department of Computer Science and Engineering, Jawaharlal Nehru University (JNU), Delhi. His research interests include peer-to-peer networks, social networks, game theory, NLP, and machine learning.

ATUL GUPTA received the Ph.D. degree from the Department of Computer Science and Engineering, Indian Institute of Technology at Kanpur (IIT Kanpur), India, in 2008. He is currently working as an Associate Professor with Indian Institute of Information Technology at Jabalpur, Jabalpur, India. His research interests include NLP application in software engineering domains, machine learning, software testing, and object-oriented software development.