Stochastic Approximation for Multi-period Simulation Optimization with Streaming Input Data

LINYUN HE, Georgia Institute of Technology, USA

UDAY V. SHANBHAG, The Pennsylvania State University, USA

EUNHYE SONG, Georgia Institute of Technology, USA

We consider a continuous-valued simulation optimization (SO) problem, where a simulator is built to optimize an expected performance measure of a real-world system while parameters of the simulator are estimated from streaming data collected periodically from the system. At each period, a new batch of data is combined with the cumulative data and the parameters are re-estimated with higher precision. The system requires the decision variable to be selected in all periods. Therefore, it is sensible for the decision-maker to update the decision variable at each period by solving a more precise SO problem with the updated parameter estimate to reduce the performance loss with respect to the target system. We define this decision-making process as the multi-period SO problem and introduce a multi-period stochastic approximation (SA) framework that generates a sequence of solutions. Two algorithms are proposed: Re-start SA (**ReSA**) reinitializes the stepsize sequence in each period, whereas Warm-start SA (**WaSA**) carefully tunes the stepsizes, taking both fewer and shorter gradient-descent steps in later periods as parameter estimates become increasingly more precise. We show that under suitable strong convexity and regularity conditions, **ReSA** and **WaSA** achieve the best possible convergence rate in expected sub-optimality either when an unbiased or a simultaneous perturbation gradient estimator is employed, while **WaSA** accrues significantly lower computational cost as the number of periods increases. In addition, we present the **regularized ReSA** which obviates the need to know the strong convexity constant and achieves the same convergence rate at the expense of additional computation.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: multi-period simulation optimization, multi-period stochastic approximation, simulation optimization under input model risk

ACM Reference Format:
Linyun He, Uday V. Shanbhag, and Eunhye Song. 2018. Stochastic Approximation for Multi-period Simulation Optimization with Streaming Input Data. *ACM Trans. Model. Comput. Simul.* 1, 1 (July 2018), 26 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

In this paper, we consider a simulation optimization (SO) problem where a high-resolution stochastic simulator is built to mimic a target system's stochastic behavior with the goal of optimizing an expected performance measure. Such a simulator is often referred to as a digital twin [22] and is continuously improved as additional data from the target

Authors' addresses: Linyun He, lhe85@gatech.edu, Georgia Institute of Technology, School of Industrial and Systems Engineering, 755 Ferst Dr. NW, Atlanta, Georgia, USA, 30332; Uday V. Shanbhag, udaybag@psu.edu, The Pennsylvania State University, Department of Industrial and Manufacturing Engineering, 201 Old Main, University Park, Pennsylvania, USA, 16802; Eunhye Song, eunhye.song@isye.gatech.edu, Georgia Institute of Technology, School of Industrial and Systems Engineering, 755 Ferst Dr NW, Atlanta, Georgia, USA, 30332.

Manuscript submitted to ACM

system are collected. Meanwhile, the decision-maker applies an SO algorithm to the simulator to find an implementable decision for the target system. Such a decision-making framework, supported by a digital twin, has been discussed in the literature in the context of supply chain [14, 37], manufacturing [26, 42], and more. We draw motivation from the following application in emergency medical service (EMS), which is revisited in Section 7 for an empirical study.

- A regional emergency medical service (EMS) provider has ambulance dispatching stations and uses simulation to support their operational decisions [38]. Their primary goal is to minimize the average response time, the time between receiving an emergency call from a patient and picking up the patient. In addition to the dispatching stations, the EMS provider also operates an additional "mobile station," where the ambulances are parked near potential sources of emergencies to minimize the average response time. The emergency call data (time, location, emergency type, etc.) are collected continuously by the EMS provider from which the spatio-temporal distribution of emergency calls is updated over time. Naturally, the location of the mobile dispatching station can also be updated as the distribution is learned more precisely.

In our problem context, the logical discrepancy between the system and the simulator is deemed negligible. We focus on the case where the parameters of the simulation input distribution function are estimated from streaming data collected from the target system as in the EMS example. We assume that the data-generating process is stationary regardless of the decision implemented in the system. Notice that in the EMS example, the implemented decision in the target system (mobile station's location) does not influence the input-generating process (distribution of emergency calls) and only affects the output performance measure (average response time).

A generic SO problem that minimizes the expected performance measure can be written as

$$\min_{\mathbf{x}\in\mathcal{X}} \mathbb{E}_{\omega}[F(\mathbf{x},\boldsymbol{\theta},\omega)], \tag{Opt($\boldsymbol{\theta}$)}$$

where $\mathcal{X} \subseteq \mathbb{R}^d$ is the set of feasible solution, F denotes the simulation output function, and $\boldsymbol{\theta}$ represents the parameter vector of the input models. We assume that 1) F is completely characterized by $\mathbf{x}$, $\boldsymbol{\theta}$, and a sequence of Uniform$(0, 1)$ random numbers represented by ω ; and 2) no analytical expression for F and that F can only be evaluated by running a simulation. The objective function in $\mathbf{x}$ is defined as the expectation of $F(\mathbf{x},\boldsymbol{\theta},\omega)$ with respect to ω (i.e. $\mathbb{E}_{\omega}\,[F(\mathbf{x},\boldsymbol{\theta},\omega)]$). If we denote the true parameter vector of the data-generating distribution in the system by $\boldsymbol{\theta}^*$, then the true optimum in the system, $\mathbf{x}^*$, is optimal for Opt$(\boldsymbol{\theta}^*)$. In general, an optimal solution of Opt$(\boldsymbol{\theta})$ for $\boldsymbol{\theta} \neq \boldsymbol{\theta}^*$ is suboptimal for Opt$(\boldsymbol{\theta}^*)$. Hence, when the unknown $\boldsymbol{\theta}^*$ is replaced with an estimate computed from finite data, there is risk of making sub-optimal decisions due to the finite-sample error in the estimate referred to as input model risk [27]. While most SO approaches accounting for input model risk assume a single batch of input data is available, [39] first incorporates streaming input data in SO; they propose a sequential ranking and selection (R&S) algorithm for a discrete SO problem where the parameter estimates for the simulator are improved from the streaming input data. However, their work only considers the case when the parameter estimator is in the form of a sample mean and a finite solution space. Our work can incorporate a general M-estimator for $\boldsymbol{\theta}^*$ and focuses on a continuous solution space. In particular, the solution procedure considered here is stochastic approximation (SA) that takes a stochastic gradient descent step at each iteration, where the stochastic gradient estimator is computed via simulations. Each iteration may require projecting the solution back to the feasible region (e.g., Euclidean projection) whose computational cost depends on the complexity of $\mathcal{X}$.

To clearly define the problem of interest, we coin the term, *multi-period* SO; Figure 1 provides a schematic of how data collection and decision-making are synchronized in multi-period SO. At the beginning of the kth period, a new batch of input data are combined with the accumulated data from the previous periods; we do not assume the batch size

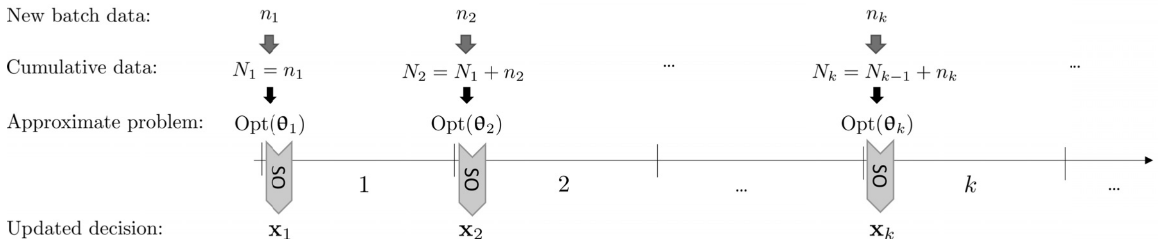


Fig. 1. The data collection and decision time line of multi-period SO problem; $\{n_k\}$ and $\{N_k\}$ respectively represent the sequence of incoming data size and cumulative data size; an SO algorithm is applied to solve $\{\text{Opt}\boldsymbol{\theta}_k\}$ and returns $\{\mathbf{x}_k\}$ at each period.

to be known a priori or can be controlled. We assume that the length of each period is much larger than the runtime of a single simulation replication. From the combined data, the kth-period estimator, $\boldsymbol{\theta}_k$, of $\boldsymbol{\theta}^*$ is updated. As k increases, the sequence of $\boldsymbol{\theta}_k$ becomes closer to $\boldsymbol{\theta}^*$ under some regularity conditions. For small k, on the other hand, $\boldsymbol{\theta}_k$ may be a poor estimate of $\boldsymbol{\theta}^*$, thus the solution to Opt($\boldsymbol{\theta}_k$) may perform poorly in the target system. One may attempt to wait until k is sufficiently large so that $\boldsymbol{\theta}_k$ has small estimation error, then solve Opt($\boldsymbol{\theta}_k$). However, the target system requires a decision to be implemented in the meantime (e.g., the mobile station's location must be determined in the EMS example). Choosing an arbitrary solution to be implemented until k is "large enough" may result in a significant loss in the system performance. Moreover, because the data streaming process is outside of our control, it is difficult to know a priori how many periods will have to pass until the desired precision is reached. Our goal is to find a sequence of solutions $\{\mathbf{x}_k\}$ such that the cumulative performance loss in the real-world system compared to $\mathbf{x}^*$ is minimal.

Our strategy is to design a multi-period SA scheme that (approximately) solves Opt($\boldsymbol{\theta}_k$) at the beginning of the kth period and implements the updated decision in the system until the next period. The key research question here is to determine how much computational effort to spend at each period so that the expected sub-optimality under $\boldsymbol{\theta}^*$ diminishes at the fastest possible rate. Hypothetically, suppose we spend infinite computational effort at each k to find the exact solution to Opt($\boldsymbol{\theta}_k$). Of course, this is unrealistic as each period is finite. Although we assume simulation runs are much faster than the data streaming process, there is a natural limit on how many replications can be performed within each period. Even if Opt($\boldsymbol{\theta}_k$) can be solved to optimality for each k, the resulting solution may be sub-optimal under $\boldsymbol{\theta}^*$, particularly for earlier periods when the estimate of $\boldsymbol{\theta}^*$ may be coarse. Nonetheless, the hypothetical scheme provides a useful benchmark for our proposed multi-period SA scheme; given $\{\boldsymbol{\theta}_k\}$, observe that the performance of any multi-period SO procedure is constrained by the rate at which the real-world $\boldsymbol{\theta}_k$ converges to $\boldsymbol{\theta}^*$. In fact, a benchmark that achieves the same rate is one that returns the optimal solution of Opt($\boldsymbol{\theta}_k$) at each k in terms of the expected sub-optimality for Opt($\boldsymbol{\theta}^*$). This motivates the design of an SA framework that spends minimal computational effort in solving Opt($\boldsymbol{\theta}_k$) at each period k while the resulting sequence of solutions provably attains the same convergence rate in expected sub-optimality as that displayed by the sequence of the optimal solutions to Opt($\boldsymbol{\theta}_k$).

Indeed, the idea of controlling the computational effort to match the precision of the approximated problem has been explored before to design an efficient sample average approximation (SAA) scheme [23, 32]. In particular, [23] studies the retrospective approximation (RA) that solves a sequence of sample-average problems constructed with increasing sample sizes with progressively tighter precision requirements (i.e., decreasing error tolerance imposed on the distance between the kth SAA problem's optimal solution and the solution returned by the algorithm) warm-starting from the previous period's solution. While philosophically multi-period SA is constructed in the same vein as RA, there are two major differences: 1) the approximation error of $\mathrm{Opt}(\boldsymbol{\theta}_k)$ is determined by not only the stochasticity in the algorithm under our control, but also the streaming input data size beyond our control; 2) the analytical expression for the simulation output function F is unknown and thus, the sample-average version of $\mathrm{Opt}(\boldsymbol{\theta})$ cannot be written down analytically or solved directly.

In this paper, our focus is predominantly on problems where $\mathbb{E}_\omega[F(\mathbf{x}, \boldsymbol{\theta}, \omega)]$ is μ-strongly convex over convex set X for any $\boldsymbol{\theta} \in \Theta$. While the convergence theory established in the paper does not immediately extend to nonconvex settings, an empirical study on a nonconvex function suggests promise (See Section 7.4). Next, we provide a brief review of the prior work in the context of such simulation optimization problems.

1.1 Literature review

There are several recent studies that consider SO under input model risk when a batch of input data is given. In [6, 27, 28], variations of R&S procedures are proposed to provide a probabilistic guarantee that the optimum of $\mathrm{Opt}(\widehat{\boldsymbol{\theta}})$, where $\widehat{\boldsymbol{\theta}}$ is an estimator of $\boldsymbol{\theta}^*$ computed from a single batch of data, is still optimal for $\mathrm{Opt}(\boldsymbol{\theta}^*)$. Such a guarantee typically relies on the requirement that $\widehat{\boldsymbol{\theta}}$ converges to $\boldsymbol{\theta}^*$ as the data size tends to infinity. However, for smaller sample sizes, the desired level of probabilistic guarantee may not be achieved by these approaches. Alternatively, [9, 12] propose distributionally robust R&S procedures that solve $\mathrm{Opt}(\boldsymbol{\theta})$ when the uncertainty set of $\boldsymbol{\theta}$ contains a finite number of candidate choices of $\boldsymbol{\theta}$.

In [7], the authors apply a Bayesian approach to model uncertainty about $\boldsymbol{\theta}^*$ by imposing a prior distribution and updating it to a posterior based on the batch input data. They adopt a modified objective function $\mathbb{E}[F(\mathbf{x}, \boldsymbol{\theta}, \omega)]$, where the expectation is taken with respect to the posterior distribution of $\boldsymbol{\theta}$ as well as stochastic simulation error induced by ω. In [24] and [36], Bayesian optimization (BO) algorithms are presented to address the same objective function as in [7] defined on a continuous solution space.

All of the aforementioned approaches focus on the case when a single batch of input data is available. Some recent research has considered the incorporation of additional data collection into the SO problem to balance the trade-off between simulation replication and input data collection [35, 40, 41]. For continuous SO, warm-start BO algorithms [25, 32] may be applied to solve a sequence of problems updated by the streaming data, although these algorithms are created for more general dynamic problems. The closest SO problem to ours appears in [21], where they adopt Bayesian input modeling and apply stochastic gradient descent to find the optimal solution for the problem where the objective function is defined by averaging both stochastic uncertainty and Bayesian posterior uncertainty.

Similar problems have been studied under the name of *misspecified optimization* in the optimization literature in which the parameters of the optimization problem are estimated from idiosyncratic data. Research on misspecified optimization and game-theoretic problems appears to be rooted in [33], [3], and [4]. More recently, [1] examines misspecified deterministic convex optimization problems while stochastic variants have been studied in [17] and [15]. These have motivated the studies on misspecified Nash equilibrium problems [18] and misspecified Markov Decision Processes [16]. The problem closest to ours is considered in [17], where learning $\boldsymbol{\theta}^*$ is cast as a stochastic convex optimization problem while $\mathrm{Opt}(\boldsymbol{\theta})$ is a stochastic convex optimization problem.

Gaps. There is a clear gap that exists in the literature designing an SA algorithm for an SO problem defined on a continuous solution space under input model risk, particularly when streaming input data are available from the system. Applying an out-of-the-box gradient descent scheme to solve the problem is faced with following challenges: (i) *Cumulative regret:* Each period's decision is implemented in the real-world system and therefore, the algorithm should be designed to return a sequence of solutions that result in small cumulative regret for the real-world problem, Opt($\boldsymbol{\theta}^*$) (See Eq. (4)). (ii) *Computational cost:* Naive approaches to reduce the regret may impose significant computational burden in computing approximate solutions of Opt($\boldsymbol{\theta}_k$). To design an efficient algorithm, information from the prior periods need to be utilized; (iii) *Knowledge of problem parameters.* Prior schemes rely on the problem parameters (e.g., strong convexity parameter) in selecting algorithm parameters, which may not be known for a typical SO problem.

1.2 Contributions

Motivated, in large part, by the gaps in the literature, this paper makes the following contributions.

(i) We define the multi-period SO problem with exogenous streaming input data and devise a multi-period SA framework that solves Opt($\boldsymbol{\theta}_k$) up to the level of error determined by the cumulative streaming data size at each period k. We propose two variants of multi-period SA: Re-start Stochastic Approximation (**ReSA**); and Warm-start Stochastic Approximation (**WaSA**). While at each period both variants warm-start from the previous period's solution, the former restarts the stepsize sequence for SA in each period, whereas the latter carefully controls the stepsize sequence for all periods as a function of the cumulative input data size.

(ii) Under a suitable strong convexity requirement on Opt($\boldsymbol{\theta}$), we show that both **ReSA** and **WaSA** achieve *the optimal convergence rate in the expected sub-optimality* given the streaming data size when combined with either an unbiased gradient estimator or the simultaneous perturbation (SP) gradient estimator [30]. For the latter, we further investigate the trade-off between increasing the simulation effort for gradient estimation vs. the number of gradient-descent steps taken within each period. We show that when the projection operation is expensive, computational effort can be significantly reduced by estimating the gradient with increasing precision, which may lead to taking fewer projection steps in each period.

(iii) We show that **WaSA** can save significant computational cost over **ReSA** by taking fewer, smaller gradient-descent steps in later periods. For instance, when the same number of data points stream in at each period and an unbiased gradient estimator is available, the cumulative number of gradient-descent steps **ReSA** takes grows quadratically in the number of periods, whereas that of **WaSA** grows linearly. The relative saving grows even more starkly when the SP gradient estimator is adopted.

(iv) Finally, we present a regularized variant of **ReSA** which does not require knowledge of the strong convexity parameter μ and proceed to show that under suitable choices of the regularization sequence, the resulting expected sub-optimality error diminishes at the same rate as **ReSA** and **WaSA** at the expense of computational effort.

In our preliminary work [29], a prior version of the multi-period SA framework is presented. However, there are several limitations of [29] that we aim to address in this paper: (i) an unbiased gradient estimator is assumed available, which may not for a general SO problem; (ii) the derivation of the upper bound on the expected sub-optimality was preliminary and one of the results was afflicted by an error; and (iii) the warm-start algorithm in [29] does not control the stepsize sequences across periods and it has proved challenging to obtain a valid upper bound on the expected sub-optimality for this algorithm once the error in (ii) is fixed; (iv) the regularized variant of **ReSA** is new.

The rest of the paper is organized as follows. In section 2, we formally state the multi-period SA problem and describe our algorithms. Section 3 summarizes the main results of this paper. Section 4 and Section 5 discuss the detailed analysis when an unbiased gradient estimator and the SP gradient estimator is employed, respectively. A regularized variant of **ReSA** is presented and analyzed in Section 6. In Section 7, we evaluate the empirical performance of our algorithms using synthetic and realistic SO problems including the EMS example described earlier. Concluding remarks are given in Section 8. All proofs of the theoretical results are included in the Supplementary Material (SM).

Notation. For arbitrary sequence $\{a_k\}$ and positive sequence $\{b_k\}$, we adopt the notation, $a_k = O(b_k)$, if there exists constant $0 < \mathcal{M} < \infty$ and $k_0 \in \mathbb{Z}$ such that $|a_k| < \mathcal{M} b_k$ for all $k \geq k_0$. We use $\|\cdot\|$ to represent the Euclidean norm.

2 PROBLEM AND ALGORITHM DESCRIPTION

In this section, we formally define the multi-period SA problem (Section 2.1) and then propose two multi-period SA algorithms to solve this problem (Section 2.2). Some background on SA is provided in SM Section **??** for completeness.

2.1 Problem statement

Suppose that there are I data-generating processes in the system, each of which streams an independent and identically distributed (i.i.d.) sequence of data. We assume that if two or more inputs are correlated, then they are collected together as vectors. Otherwise, all I inputs are independent of each other. The parametric family of each input distribution is assumed to be known, however, its true parameter(s) must be estimated from the data. We aggregate the true parameters of all I inputs within the vector $\boldsymbol{\theta}^* \in \mathbb{R}^{I'}$, where $I' \geq I$; see Section **??** of SM for further discussion.

Let $\mathsf{n}_k^1, \mathsf{n}_k^2, \ldots, \mathsf{n}_k^I$ denote the number of observations obtained from I real-world input distributions at the kth period and $n_k = \sum_{i=1}^{I} \mathsf{n}_k^i$, i.e., n_k is the sum of incoming data size at the kth period from all I input-generating processes. We assume that n_k^i / n_k converges to constant ρ_i with probability 1 as $k \to \infty$ for each $i = 1, 2, \ldots, I$. Without loss of generality, it can be assumed that $n_k > 0$, because any period in which no data are collected can be merged with the next period. The cumulative average number of observations by the kth period is denoted by N_k, i.e. $N_k \triangleq \sum_{\ell=1}^{k} n_\ell$. Let $\mathcal{Z}_k$ be the collection of all input data observed from the system up to the kth period. Given $\mathcal{Z}_k$, $\boldsymbol{\theta}^*$ is approximated by its estimator $\boldsymbol{\theta}_k$. Our scheme allows $\boldsymbol{\theta}_k$ to be any parametric estimator that satisfies

$$\boldsymbol{\theta}_k \xrightarrow{a.s.} \boldsymbol{\theta}^* \quad \text{and} \quad \mathbb{E}[\|\boldsymbol{\theta}_k - \boldsymbol{\theta}^*\|^2] = O\left(N_k^{-1}\right), \tag{1}$$

where a.s. denotes almost sure convergence. An example of such $\boldsymbol{\theta}_k$ is an M-estimator defined as a solution to

$$\min_{\boldsymbol{\theta} \in \Theta} \mathcal{L}(\boldsymbol{\theta} \mid \mathcal{Z}_k), \tag{2}$$

where $\Theta \subseteq \mathbb{R}^{I'}$ for $I' \geq I$ is a feasible set for $\boldsymbol{\theta}$ and $\mathcal{L}(\cdot \mid \mathcal{Z}_k)$ is a sample loss function given cumulative observations $\mathcal{Z}_k$. For instance, if $\mathcal{L}(\cdot \mid \mathcal{Z}_k)$ is chosen to be the negative log-likelihood function of $\mathcal{Z}_k$, then the resulting $\boldsymbol{\theta}_k$ is a maximum likelihood estimator; see Section **??** of SM for the exact form of $\mathcal{L}$ in this case. For this choice of $\mathcal{L}$, assuming that (2) can be solved to optimality, $\boldsymbol{\theta}_k$ satisfies (1) under some regularity conditions presented in Section **??**.

We introduce the following notation for the conditional mean of the simulation output at fixed $\mathbf{x}$ given $\boldsymbol{\theta}$:

$$f(\mathbf{x}, \boldsymbol{\theta}) \triangleq \mathbb{E}_\omega[F(\mathbf{x}, \boldsymbol{\theta}, \omega)]. \tag{3}$$

Recall that $\mathbb{E}_\omega$ indicates the expectation is taken with respect to ω. Note that (3) allows $\mathbf{x}$ to be either a deterministic feasible solution or a random solution returned by an SO algorithm.

Recall that Opt($\boldsymbol{\theta}^*$) is defined on the d-dimensional continuous feasible solution space, $\mathcal{X}$, and its solution is denoted by $\mathbf{x}^*$. Let $\mathbf{x}_k^*$ represent the optimal solution to Opt($\boldsymbol{\theta}_k$), i.e, $\mathbf{x}_k^* = \arg\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}, \boldsymbol{\theta}_k)$. Since $\boldsymbol{\theta}_k$ is progressively getting closer to $\boldsymbol{\theta}^*$ as k increases, we may expect $\mathbf{x}_k^*$ to get closer to $\mathbf{x}^*$ under some suitable smoothness assumption on f.

Consider a multi-period SA scheme that takes M_k stochastic gradient-descent steps to solve Opt($\boldsymbol{\theta}_k$) and returns $\mathbf{x}_k$ as its (approximate) solution at each period k. Here, the stochastic gradient estimator is computed from simulation replications; we elaborate on the types of stochastic gradient estimators considered in our analyses in Sections 4.1

Algorithm 1 (**ReSA**). Re-start multi-period SA

```
Given $\mathbf{x}_0$, $\gamma_0$, $p$, $\{N_k\}$; Set $k := 1$;
[1] Compute $\boldsymbol{\theta}_k$ using $N_k$ samples; Set $\mathbf{x}_{k,1} := \mathbf{x}_{k-1}$ and $M_k := \max\left\{1, \left\lceil N_k^{1/p} \right\rceil\right\}$;
[2] $\mathbf{x}_{k,j+1} := \Pi_{\mathcal{X}}\left[\mathbf{x}_{k,j} - \gamma_{k,j} G(\mathbf{x}_{k,j}, \boldsymbol{\theta}_k, \omega_{k,j})\right]$ given $\gamma_{k,j} = \frac{\gamma_0}{j}$ for $j = 1, \cdots, M_k$;
[3] $\mathbf{x}_k := \mathbf{x}_{k,M_k+1}$; $k := k + 1$ and go to [1];
```

and 5.1. Since M_k is finite, $\mathbf{x}_k$ is subject to stochastic error, i.e., $\mathbf{x}_k \neq \mathbf{x}_k^*$ in general. The performance loss of implementing $\mathbf{x}_k$ in the system at the kth period can be measured by the optimality gap between $\mathbf{x}_k$ and $\mathbf{x}^*$ under the target system problem, Opt($\boldsymbol{\theta}^*$), as $\boldsymbol{\theta}^*$ characterizes the system for which the sequence of decisions, $\{\mathbf{x}_k\}$, is implemented. To evaluate the performance loss over K periods, we consider the following expected cumulative sub-optimality (regret) of the sequence of solutions, $\{\mathbf{x}_k\}$, for Opt($\boldsymbol{\theta}^*$):

$$\sum\nolimits_{k=1}^{K} \mathbb{E}\left[f(\mathbf{x}_k, \boldsymbol{\theta}^*) - f(\mathbf{x}^*, \boldsymbol{\theta}^*)\right], \tag{4}$$

where the expectation is taken with respect to the sampling error in $\boldsymbol{\theta}_k$ caused by finiteness of N_k as well as the stochastic variability in computing $\mathbf{x}_k$. We emphasize that the kth-period expected sub-optimality in (4) is not $\mathbb{E}[f(\mathbf{x}_k, \boldsymbol{\theta}_k) - f(\mathbf{x}^*, \boldsymbol{\theta}^*)]$; this is because we implement $\mathbf{x}_k$ in the target system whose true, but unknown parameter, is $\boldsymbol{\theta}^*$.

As mentioned in Section 1, the cumulative expected sub-optimality of $\{\mathbf{x}_k^*\}$ provides a benchmark for any multi-period SA scheme. Hypothetically, suppose simulation is instantaneous so that for each Opt($\boldsymbol{\theta}_k$), we can take infinitely many gradient-descent steps to find $\mathbf{x}_k^*$. Still, $\mathbb{E}[f(\mathbf{x}_k^*, \boldsymbol{\theta}^*) - f(\mathbf{x}^*, \boldsymbol{\theta}^*)] \geq 0$ as $\boldsymbol{\theta}_k \neq \boldsymbol{\theta}^*$ in general due to the sampling error in $\boldsymbol{\theta}_k$. Therefore, even if the kth period's SA scheme is stopped after $M_k < \infty$ gradient-descent steps, as long as

$$\mathbb{E}[f(\mathbf{x}_k^*, \boldsymbol{\theta}^*) - f(\mathbf{x}^*, \boldsymbol{\theta}^*)] \approx \mathbb{E}[f(\mathbf{x}_k, \boldsymbol{\theta}^*) - f(\mathbf{x}^*, \boldsymbol{\theta}^*)] \tag{5}$$

for each k, the expected cumulative sub-optimality of $\{\mathbf{x}_k\}$ would be similar to that of $\{\mathbf{x}_k^*\}$. Thus, our goal lies in developing a multi-period SA algorithm such that both expected sub-optimality terms in (5) have the same convergence rate in N_k by carefully selecting M_k as a function of N_k at each k.

Remark: Instead of $\mathbf{x}_k$, one may consider adopting $\bar{\mathbf{x}}_k = \arg\min_{\mathbf{x}\in\mathcal{X}} \mathbb{E}[f(\mathbf{x}, \boldsymbol{\theta}_k)]$, where the expectation is with respect to the (approximate) sampling distribution of $\boldsymbol{\theta}_k$. In Appendix **??**, we show that under assumptions similar to those made in Section 3, $\{\bar{\mathbf{x}}_k\}$ and $\{\mathbf{x}_k\}$ have the same rates of convergence in expected sub-optimality and discuss computational advantage of $\mathbf{x}_k$ over $\bar{\mathbf{x}}_k$.

2.2 Algorithm definition

We propose two variants of multi-period SA. The first is the *re-start multi-period SA* (**ReSA**) scheme presented in Algorithm 1. **ReSA** initializes (restarts) the stepsize sequence for gradient descent at each period while adopting the previous period's solution as the initial solution. In the kth period, **ReSA** solves Opt($\boldsymbol{\theta}_k$) by taking M_k projected gradient-descent steps starting from $\mathbf{x}_{k-1}$. For the first period, we assume $\mathbf{x}_0$ is randomly selected in $\mathcal{X}$ in our analyses. Let $\mathbf{x}_{k,j}$, $j = 1, 2, \ldots, M_k$ denote the sequence of solutions returned by the algorithm in the kth period.

Note that $G(\mathbf{x}_{k,j}, \boldsymbol{\theta}_k, \omega_{k,j})$ in Step [2] is a stochastic estimator of $\nabla_{\mathbf{x}} f(\mathbf{x}_{k,j}, \boldsymbol{\theta}_k)$. Depending on the choice of G, $\omega_{k,j}$ may be a sequence of Uniform $(0, 1)$ random numbers or a collection of such sequences if G requires multiple replications to compute. In addition, $\Pi_{\mathcal{X}}(u)$ represents the Euclidean projection of vector $u \in \mathbb{R}^d$ onto $\mathcal{X}$ while γ_0 denotes the constant of the stepsize sequence. Note that we use the parameter p to define the $\{M_k\}$ sequence, where p

Algorithm 2 (WaSA). Warm-start multi-period SA

```
Given $\mathbf{x}_0$, $\gamma_0$, $\widetilde{\gamma}_0$, $p$, $\{N_k\}$; Choose $0 < \lambda < 1/p$; Set $k := 1$;
[1] Run ReSA with $\gamma_0$ and $\mathbf{x}_0$ for $k = 1$ to obtain $\mathbf{x}_1$; Update $k := 2$;
[2] Compute $\boldsymbol{\theta}_k$ using $N_k$ samples; $M_k := \left\lceil N_k^{1/p} - N_{k-1}^{\lambda} \right\rceil$; Set $\mathbf{x}_{k,1} := \mathbf{x}_{k-1}$;
[3] $\mathbf{x}_{k,j+1} := \Pi_X \left[ \mathbf{x}_{k,j} - \gamma_{k,j} G(\mathbf{x}_{k,j}, \boldsymbol{\theta}_k, \omega_{k,j}) \right]$ given $\gamma_{k,j} = \frac{\widetilde{\gamma}_0}{N_{k-1}^{\lambda} + j - 1}$ for $j = 1, \cdots, M_k$;
[4] $\mathbf{x}_k := \mathbf{x}_{k,M_k+1}$; $k := k + 1$ and go to [2];
```

is chosen so that $\mathbb{E}\left[\, \|\mathbf{x}_{1,j} - \mathbf{x}_1^*\|^2 \middle| \boldsymbol{\theta}_1 \right] = O(j^{-p})$, which can be achieved by choosing M_1 and $\{\gamma_{1,j}\}$ appropriately. The value of p depends on the gradient estimator. In the remainder of the paper, we refer to $O(j^{-p})$ as the *single-period MSE convergence rate* to differentiate it from the convergence rate of the expected sub-optimality, $\mathbb{E}[f(\mathbf{x}_k, \boldsymbol{\theta}^*) - f(\mathbf{x}^*, \boldsymbol{\theta}^*)]$, which we simply refer to as the *convergence rate*. Notice that in Step [2], $\gamma_{k,1}$ is reset to γ_0 for each k. The algorithm parameters are left unspecified here because their choices depend on the properties of $\text{Opt}(\boldsymbol{\theta})$ and G; these are clarified in Sections 4–5 with corresponding assumptions on $\text{Opt}(\boldsymbol{\theta})$ and G.

ReSA is a generalization of the scheme adopted by [17], where a single projected gradient-descent step was taken at each period, i.e. $M_k = 1$ for every $k \geq 1$. **ReSA** sets M_k to be a function of N_k at each k. The choice of M_k in **ReSA** ensures that we spend just enough computational effort in the kth period for $\{\mathbf{x}_k\}$ to achieve the same convergence rate as $\{\mathbf{x}_k^*\}$, given the streaming data sequence. In fact, the choice of $\{M_k\}$ in **ReSA** guarantees that the convergence rate matches the rate at which $\boldsymbol{\theta}_k \to \boldsymbol{\theta}^*$ even when the initial solution to the kth problem, $\mathbf{x}_{k,1}$, is chosen randomly in $\mathcal{X}$ under appropriate assumptions made in Sections 3–5. Since $\boldsymbol{\theta}_k \xrightarrow[k\to\infty]{a.s.} \boldsymbol{\theta}^*$ and $\mathbf{x}_{k-1}^*$ and $\mathbf{x}_k^*$ become closer as k increases, one may expect that taking $\mathbf{x}_{k-1}$ close to $\mathbf{x}_{k-1}^*$ as the initial solution of the kth period would improve the computational efficiency of the algorithm by letting us take fewer and smaller SA (gradient-descent) steps while achieving the same convergence rate. Based on this intuition, we propose the *warm-start multi-period SA* (**WaSA**) scheme in Algorithm 2.

The constant, λ, can be any value in $(0, 1/p)$; the closer λ is to $1/p$, the fewer gradient-descent steps **WaSA** takes in each period. The choices for the parameters are discussed in Sections 4–5 along with the assumptions on $\text{Opt}(\boldsymbol{\theta})$ and G.

Observe from Step [1] of Algorithm 2 that **ReSA** and **WaSA** make identical progress for $k = 1$. For $k \geq 2$, **WaSA** displays two key distinctions from **ReSA**:

(a) By setting the jth stepsize at period k as $\gamma_{k,j} = \tilde{\gamma}_0/(N_{k-1}^{\lambda} + j - 1)$, **WaSA** takes increasingly smaller steps within each period as k increases.

(b) By choosing M_k smaller than its corresponding value in **ReSA**, **WaSA** takes increasingly fewer SA steps than **ReSA** as k increases.

Hence, as k increases and more input data are accumulated, **WaSA** takes smaller and also fewer gradient-descent steps recognizing that it is in the vicinity of $\mathbf{x}^*$.

To summarize, both **ReSA** and **WaSA** warm-start from the previous period's implemented solution. The difference lies in the choices for $\{\gamma_{k,j}\}$ as well as $\{M_k\}$. While both algorithms are designed so that $\mathbb{E}[f(\mathbf{x}_k, \boldsymbol{\theta}^*) - f(\mathbf{x}^*, \boldsymbol{\theta}^*)]$ achieves the same convergence rate as $\mathbb{E}[f(\mathbf{x}_k^*, \boldsymbol{\theta}^*) - f(\mathbf{x}^*, \boldsymbol{\theta}^*)]$, employing **WaSA** may lead to significant computational savings by taking fewer and more carefully specified gradient-descent steps.

3 OVERVIEW OF MAIN RESULTS

In this section, we summarize the main theoretical properties of **ReSA** and **WaSA**. To facilitate the discussion, we first provide a set of conditions on $\text{Opt}(\boldsymbol{\theta})$.

Assumption 1. *The feasible solution set, $\mathcal{X} \subseteq \mathbb{R}^d$, is closed, convex, and nonempty and $\Theta \subseteq \mathbb{R}^{I'}$ is nonempty and compact. For each $\boldsymbol{\theta} \in \Theta$, $f(\cdot, \boldsymbol{\theta})$ is μ-strongly convex and continuously differentiable on an open set containing $\mathcal{X}$.*

Assumption 2. *There exist $L_{\mathcal{X}} > 0$ and $L_\Theta > 0$ such that $\|\nabla_{\mathbf{x}} f(\mathbf{x}, \boldsymbol{\theta}) - \nabla_{\mathbf{x}} f(\mathbf{y}, \boldsymbol{\theta})\| \le L_{\mathcal{X}} \|\mathbf{x} - \mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ and for all $\boldsymbol{\theta} \in \Theta$ and $\|\nabla_{\mathbf{x}} f(\mathbf{x}, \boldsymbol{\theta}_1) - \nabla_{\mathbf{x}} f(\mathbf{x}, \boldsymbol{\theta}_2)\| \le L_\Theta \|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|$ for all $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \Theta$ and for all $\mathbf{x} \in \mathcal{X}$.*

Assumption 3. *The optimal solution of $\mathrm{Opt}(\boldsymbol{\theta}^*)$, denoted by $\mathbf{x}^*$, is an interior point of $\mathcal{X}$.*

Note that our algorithms do not not require $L_{\mathcal{X}}$ and L_Θ to be known; they only need to exist. On the other hand, we assume the strong convexity parameter, μ, is known in Sections 4–5. Smooth function h with domain $\mathcal{X} \subseteq \mathbb{R}^d$ is said to be μ-strongly convex, if for any $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, $(\nabla h(\mathbf{x}) - \nabla h(\mathbf{x}'))^\top (\mathbf{x} - \mathbf{x}') \ge \mu \|\mathbf{x} - \mathbf{x}'\|^2$ for some $\mu > 0$. Depending on the problem context, this may or may not be a restrictive assumption. For instance, Section 7.2 features a stochastic activity network problem for which the objective function includes a deterministic cost function with a known strong convexity parameter. On the other hand, in the EMS example in Section 7.3 we cannot analytically confirm that the problem is strongly convex nor can we derive the value of μ even if the problem is strongly convex. In Section 6, we investigate a regularized version of **ReSA** (**r-ReSA**) scheme that does not require known μ; we further elaborate on this scheme at the end of this section.

We consider two types of stochastic gradient estimator G: a generic unbiased gradient estimator and the simultaneous perturbation (SP) gradient estimator. The latter is biased in general [30]; to balance its bias and variance, we introduce an additional algorithm parameter $s_{k,j}$ when the SP gradient estimator is adopted to control the number of simulation replications made to compute G at the jth SA step within the kth period.

For both **ReSA** and **WaSA**, the upper bound on the expected sub-optimality at the kth period is determined by how close $\mathbf{x}_k$ and $\boldsymbol{\theta}_k$ are to $\mathbf{x}^*$ and $\boldsymbol{\theta}^*$, respectively. The following lemma connects these two pieces together to provide an upper bound on $\mathbb{E}[f(\mathbf{x}_k, \boldsymbol{\theta}^*) - f(\mathbf{x}^*, \boldsymbol{\theta}^*)]$ for any generic multi-period SO framework that returns the sequence of solutions, $\{\mathbf{x}_k\}$, at each period.

Lemma 3.1. *Suppose Assumptions 1–3 hold. Consider the sequences $\{\mathbf{x}_k\}$ and $\{\boldsymbol{\theta}_k\}$ generated by any multi-period SO framework. Then, the following holds for any $k \ge 1$:*

$$\mathbb{E}[f(\mathbf{x}_k, \boldsymbol{\theta}^*) - f(\mathbf{x}^*, \boldsymbol{\theta}^*)] \le L_{\mathcal{X}} \mathbb{E}[\|\mathbf{x}_k - \mathbf{x}_k^*\|^2] + \frac{L_{\mathcal{X}} L_\Theta^2}{\mu^2} \mathbb{E}[\|\boldsymbol{\theta}_k - \boldsymbol{\theta}^*\|^2]. \tag{6}$$

Notice that when $\mathbf{x}_k = \mathbf{x}_k^*$, the first term of the upper bound in (6) vanishes. Hence, the expected sub-optimality of $\{\mathbf{x}_k^*\}$ converges at the same rate as $\mathbb{E}[\|\boldsymbol{\theta}_k - \boldsymbol{\theta}^*\|^2]$. Lemma 3.1 provides guidance on how accurately $\mathrm{Opt}(\boldsymbol{\theta}_k)$ needs to be solved at each k to achieve the best possible convergence rate. Both **ReSA** and **WaSA** can control $\mathbb{E}[\|\mathbf{x}_k - \mathbf{x}_k^*\|^2]$ by choosing M_k and $\gamma_{k,j}$ appropriately. Since $\mathbb{E}[\|\boldsymbol{\theta}_k - \boldsymbol{\theta}^*\|^2] = O(N_k^{-1})$, the best convergence rate can be obtained when $\mathbb{E}[\|\mathbf{x}_k - \mathbf{x}_k^*\|^2] = O(N_k^{-1})$; there is no reason to put additional effort to make $\mathbb{E}[\|\mathbf{x}_k - \mathbf{x}_k^*\|^2]$ decay at a faster rate. Note that we focus on the convergence rate rather than minimizing the upper bound in (6) since $L_{\mathcal{X}}$ and L_Θ are unknown.

Table 1 summarizes the convergence rates of **ReSA** and **WaSA**; for both types of gradient estimators, both algorithms achieve the best-possible rate of $O(N_K^{-1})$. Table 1 also presents how the cumulative number of SA steps, $\sum_{k=1}^{K} M_k$, spent by **ReSA** and **WaSA** grows at each period K for $K \ge 1$ to achieve this convergence rate. Recall from Algorithms 1–2, the choice of M_k depends on the exponent of the single-period MSE convergence rate, $O(j^{-p})$. The smaller p is, the more SA steps are required at each period to achieve the best convergence rate. When G is an unbiased gradient estimator, we have $p = 1$. When the SP gradient estimator is adopted, choosing $s_{k,j}$ to be a constant leads to $p = 2/3$. Increasing $s_{k,j}$ as a polynomial in j, p can be pushed to 1. The last two columns of Table 1 show the growth rate of the

Table 1. The convergence rates of expected sub-optimality, cumulative number of SA steps, and cumulative simulation effort of **ReSA** and **WaSA** at the end of the Kth period under different sets of assumptions. The rates marked with $*$ can be achieved when λ is pushed to its upper bound.

Gradient estimator	Convergence rate		Cumulative number of SA steps		Cumulative simulation effort	
	ReSA	**WaSA**	**ReSA**	**WaSA**	**ReSA**	**WaSA**
Unbiased ($p = 1$)	$O(N_K^{-1})$	$O(N_K^{-1})$	$O\left(\sum_{k=1}^K N_k\right)$	$O(N_K)^*$	$O\left(\sum_{k=1}^K N_k\right)$	$O(N_K)^*$
SP ($2/3 \le p \le 1$)	$O(N_K^{-1})$	$O(N_K^{-1})$	$O\left(\sum_{k=1}^K N_k^{1/p}\right)$	$O\left(N_K^{1/p}\right)^*$	$O\left(\sum_{k=1}^K N_k^{3/2}\right)$	$O\left(N_K^{3/2}\right)^*$

cumulative simulation effort **ReSA** and **WaSA** respectively spend on estimating the gradients up to period K. Notice that the growth rate of the cumulative simulation effort does not depend on p for the SP gradient estimator, whereas the cumulative number of SA steps does. Given that Opt($\boldsymbol{\theta}$) is a constrained optimization problem, each SA step may require a Euclidean projection to ensure feasibility, which can be computationally expensive when $\mathcal{X}$ is a complex set. In this case, it is sensible to push p closer to 1 so that fewer projections are required.

Recall that in **WaSA**, parameter λ determines the computational cost; the closer λ is to $1/p$, the more computational saving is achieved. When $\lambda = 0$, **WaSA** essentially reduces to **ReSA**. We note that Table 1 provides the upper bounds (marked with '$*$') on the smallest-possible cumulative SA steps and simulation effort that **WaSA** can achieve when λ is pushed to $1/p$, which clearly demonstrates significant computational benefits of **WaSA** over **ReSA**. Notice from Table 1 that **WaSA** achieves the same convergence rate in terms of expected sub-optimality regardless of the choice of λ, which favors λ close to $1/p$.

When μ is unknown, inspired by the classical theory of Tikhonov regularization [34], we propose regularized-**ReSA** (**r-ReSA**) in Section 6. We convert Opt($\boldsymbol{\theta}_k$) into a regularized problem with known convexity parameter μ_k and solve it instead of Opt($\boldsymbol{\theta}_k$) to obtain $\mathbf{x}_k$. We show that by choosing $\{\mu_k\}$ and the algorithm parameters carefully, the resulting expected sub-optimality diminishes at the rate of $O(N_k^{-1})$. However, **r-ReSA** requires a larger number of SA steps (and simulation effort), which is the price we pay for not knowing μ. See Section 6 for details.

4 MULTI-PERIOD SA FRAMEWORK WITH UNBIASED GRADIENT ESTIMATOR

In this section, we analyze the theoretical properties of **ReSA** and **WaSA** when unbiased estimator G of $\nabla_{\mathbf{x}} f(\mathbf{x}, \boldsymbol{\theta})$ is available. While the analysis in this section is applicable to general unbiased gradient estimator G, we discuss an example of a simulation-based unbiased gradient estimator in Section 4.1. In Section 4.2, we analyze the expected sub-optimality and computational costs of **ReSA** and **WaSA**.

4.1 Infinitesimal perturbation analysis gradient estimator

The infinitesimal perturbation analysis (IPA) gradient estimator is a well-studied unbiased gradient estimator [10]. Recall that in our notation, the simulation output, $F(\mathbf{x}, \boldsymbol{\theta}, \omega)$, is a function of $\mathbf{x}$ and $\boldsymbol{\theta}$ as well as a sequence of U(0, 1) random numbers ω that drives the stochasticity in simulation. For simplicity, consider the case where each simulation run requires a single random number, i.e., $\omega \sim U(0, 1)$. Given $\mathbf{x}$ and $\boldsymbol{\theta}$, we can write $f(\mathbf{x}, \boldsymbol{\theta}) \triangleq \mathbb{E}_\omega[F(\mathbf{x}, \boldsymbol{\theta}, \omega)] = \int_0^1 F(\mathbf{x}, \boldsymbol{\theta}, \omega) d\omega$. If the exchange of derivative with respect to $\mathbf{x}$ and the integral is allowed (see [10] for the conditions under which the exchange operation is allowed), then we have

$$\nabla_{\mathbf{x}} \mathbb{E}_\omega[F(\mathbf{x}, \boldsymbol{\theta}, \omega)] = \int_0^1 \nabla_{\mathbf{x}} F(\mathbf{x}, \boldsymbol{\theta}, \omega) d\omega. \tag{7}$$

Therefore, we have an unbiased gradient estimator of $\nabla_{\mathbf{x}} f(\mathbf{x}, \boldsymbol{\theta})$, $G = \nabla_{\mathbf{x}} F(\mathbf{x}, \boldsymbol{\theta}, \omega)$. A caveat here is that $\nabla_{\mathbf{x}} F(\mathbf{x}, \boldsymbol{\theta}, \omega)$ must be computable given ω. If $F(\mathbf{x}, \boldsymbol{\theta}, \omega)$ can be written as a function of some other intermediate random variable generated by transforming ω, then the chain rule can be applied to find $\nabla_{\mathbf{x}} F(\mathbf{x}, \boldsymbol{\theta}, \omega)$. This may be difficult, however, if the simulation logic is complex. In such a case, the SP gradient estimator discussed in Section 5.1 can be applied. In Section 7.2, we present a numerical example in which the IPA gradient estimator can be computed. Since G may be obtained from a single simulation replication, the number of SA steps taken at each period equals the simulation effort at each period. Thus, we do not differentiate between these two measures of computational effort in Section 4.2.

4.2 Rate analysis

We start by making the following additional assumptions on $\text{Opt}(\boldsymbol{\theta})$ as well as G.

Assumption 4. *The feasible solution set, $\mathcal{X} \in \mathbb{R}^d$, is convex, compact, and has a nonempty interior. Further, there exists $C_{\mathcal{X}} > 0$ such that $\|\nabla_{\mathbf{x}} f(\mathbf{x}, \boldsymbol{\theta})\| \le C_{\mathcal{X}}$ for all $\mathbf{x} \in \mathcal{X}$ and $\boldsymbol{\theta} \in \Theta$.*

Assumption 5. *For all $\mathbf{x} \in \mathcal{X}$ and $\boldsymbol{\theta} \in \Theta$, $\mathbb{E}_{\omega}[G(\mathbf{x}, \boldsymbol{\theta}, \omega) - \nabla_{\mathbf{x}} f(\mathbf{x}, \boldsymbol{\theta})] = 0$. Moreover, there exists $0 < \nu < \infty$ such that $\mathbb{E}_{\omega}[\|G(\mathbf{x}, \boldsymbol{\theta}, \omega) - \nabla_{\mathbf{x}} f(\mathbf{x}, \boldsymbol{\theta})\|^2] \le \nu^2$ for all $\mathbf{x} \in \mathcal{X}$ and $\boldsymbol{\theta} \in \Theta$.*

Under Assumptions 4 and 5, $\mathbb{E}_{\omega}\left[\|G(\mathbf{x}, \boldsymbol{\theta}, \omega)\|^2\right] \le 2\mathbb{E}_{\omega}\left[\|G(\mathbf{x}, \boldsymbol{\theta}, \omega) - \nabla_{\mathbf{x}} f(\mathbf{x}, \boldsymbol{\theta})\|^2\right] + 2\|\nabla_{\mathbf{x}} f(\mathbf{x}, \boldsymbol{\theta})\|^2 \le 2(\nu^2 + C_{\mathcal{X}}^2)$ for all $\mathbf{x} \in \mathcal{X}$ and $\boldsymbol{\theta} \in \Theta$. In the remainder of the paper, we adopt $C^2 \triangleq 2(\nu^2 + C_{\mathcal{X}}^2)$. We explicitly specify the values of the parameters to be adopted in **ReSA** and **WaSA** when Assumptions 1–5 are satisfied as follows.

Definition 4.1. For both **ReSA** and **WaSA** let $p = 1$ and $\gamma_0 = 1/\mu$. For **WaSA**, let $\tilde{\gamma}_0 = 1/\mu$ and choose $0 < \lambda < 1$.

Recall that p in both algorithms should be chosen to match the exponent of the MSE convergence rate of $\mathbf{x}_1$, i.e., $\mathbb{E}\left[\|\mathbf{x}_{1,j} - \mathbf{x}_1^*\|^2 \middle| \boldsymbol{\theta}_1\right] \le O(j^{-p})$. The following lemma shows that under Assumptions 1–5, p is indeed 1 when the algorithm parameters are chosen as in Definition 4.1.

Lemma 4.2 (Single-period MSE convergence rate). *Suppose Assumptions 1, 2, 4, and 5 hold and the algorithm parameters are chosen as in Definition 4.1. Consider $\{\mathbf{x}_k\}$ generated by **ReSA**. Then, for $k \ge 1$ and $1 \le j \le M_k$, $\mathbb{E}[\|\mathbf{x}_{k,j} - \mathbf{x}_k^*\|^2|\boldsymbol{\theta}_k] \le \max\left\{\mathbb{E}[\|\mathbf{x}_{k-1} - \mathbf{x}_k^*\|^2|\boldsymbol{\theta}_k], \frac{2C^2}{\mu^2}\right\} j^{-1}$ a.s. Moreover, when $j \ge 3$, $\mathbb{E}[\|\mathbf{x}_{k,j} - \mathbf{x}_k^*\|^2|\boldsymbol{\theta}_k] \le \frac{C^2}{\mu^2} j^{-1}$ a.s.*

Notice that $\mathbb{E}[\|\mathbf{x}_{k-1} - \mathbf{x}_k^*\|^2|\boldsymbol{\theta}_k]$ appears in the bound on $\mathbb{E}[\|\mathbf{x}_{k,j} - \mathbf{x}_k^*\|^2|\boldsymbol{\theta}_k]$ only for $1 \le j \le 2$. For $j \ge 3$, we have a tighter bound that does not depend on $\mathbb{E}[\|\mathbf{x}_{k-1} - \mathbf{x}_k^*\|^2|\boldsymbol{\theta}_k]$. In other words, even if $\mathbf{x}_{k,1}$ is chosen to be an arbitrary solution in $\mathcal{X}$, the same bound holds. This reveals that the standard SA analysis conducted in Lemma 4.2 provides little leverage on a better initial solution to obtain a tighter upper bound on the convergence rate. From Lemma 4.2, we show the following theorem stating that the convergence rate and the complexity of the cumulative number of SA steps of **ReSA** are as stated in Table 1.

Theorem 4.3 (**ReSA** with unbiased G). *Suppose Assumptions 1–5 hold and the algorithm parameters are chosen as in Definition 4.1. Then, the following hold for $\{\mathbf{x}_k\}$ generated by **ReSA**.*

(i) There exists finite $U > 0$ such that for $k \ge 1$, $\mathbb{E}\left[f(\mathbf{x}_k, \boldsymbol{\theta}^*) - f(\mathbf{x}^*, \boldsymbol{\theta}^*)\right] \le \frac{L_{\mathcal{X}}}{\mu^2}(L_{\Theta}^2 U + C^2)N_k^{-1}$.

(ii) Given $K > 0$, the cumulative number of SA steps at the end of the Kth period is $\sum_{k=1}^{K} N_k$.

Theorem 4.3 implies that **ReSA** indeed achieves the optimal convergence rate of $O(N_k^{-1})$. Although this is reassuring, the number of SA steps, M_k, taken at each iteration is tied to N_k and keeps increasing as more streaming data accumulate.

In **WaSA**, not only do we choose $\mathbf{x}_{k,1} = \mathbf{x}_{k-1}$, but also select $\gamma_{k,j}$ to be a decreasing function in k for each j, i.e. correspondingly smaller steps are taken as k grows. The core idea is the following: since each period starts with an increasingly better initial solution (in terms of proximity to $\mathbf{x}^*$), we may take fewer and smaller SA steps at each period. However, exactly how to control $\gamma_{k,j}$ so that the benefit of selecting $\mathbf{x}_{k,1} = \mathbf{x}_{k-1}$ is reflected in the upper bound on the convergence rate of $\mathbb{E}[\|\mathbf{x}_k - \mathbf{x}_k^*\|^2]$ is not a trivial question. The following theorem shows that with our choice of $\gamma_{k,j}$, **WaSA** indeed achieves the optimal convergence rate, $O(N_k^{-1})$, with much reduced computational cost than **ReSA**. In particular, as λ tends to 1, the cumulative number of SA steps after k periods tends to $O(N_k)$.

Theorem 4.4 (**WaSA** with unbiased G). *Suppose Assumptions 1–5 hold and the algorithm parameters are chosen as in Definition 4.1. Then, the following hold for $\{\mathbf{x}_k\}$ generated by* ***WaSA****.*

(i) There exist finite $U > 0$ and sequence $\{Q'_k\}$ such that for $k \geq 1$, $Q'_k \xrightarrow{k\to\infty} C^2/\mu^2$ and

$$\mathbb{E}\left[f(\mathbf{x}_k, \boldsymbol{\theta}^*) - f(\mathbf{x}^*, \boldsymbol{\theta}^*)\right] \leq L_{\mathcal{X}} \left(UL_\Theta^2/\mu^2 + Q'_k\right) N_k^{-1}.$$

(ii) Given $K \geq 2$, there exists $0 < \lambda < 1$ sufficiently close to 1 such that the cumulative number of SA steps at the end of the Kth period is bounded as $\sum_{k=1}^K M_k \leq N_K + K - 1$.

Observe that the upper bound on the expected sub-optimality of **WaSA** converges to that of **ReSA** as k increases, because $Q'_k \to C^2/\mu^2$. Hence, as more streaming data accumulate, **WaSA** returns a solution essentially as accurate as that from **ReSA** at much reduced computational cost. Nevertheless, both algorithms require increasing computational effort as k increases, albeit M_k increases significantly slower for **WaSA** when λ is close to 1. The closer λ is to 1, the longer it takes Q'_k to converge to C^2/μ^2, implying that the upper bound on the expected sub-optimality may be inflated for earlier periods. However, λ cannot be equal to 1, which makes the $\{Q'_k\}$ sequence diverge. On the other hand, if λ is close to zero, **WaSA** essentially reduces to **ReSA**. In practice, it often suffices to obtain a solution close to $\mathbf{x}^*$. We refer to random $\mathbf{x}_\epsilon \in \mathcal{X}$ satisfying

$$\mathbb{E}\left[f(\mathbf{x}_\epsilon, \boldsymbol{\theta}^*) - f(\mathbf{x}^*, \boldsymbol{\theta}^*)\right] \leq \epsilon$$

as an ϵ-solution. The next corollary establishes the computational complexity for both algorithms to obtain an ϵ-solution assuming $n_k = \lceil k^a \rceil$ for $a \geq 0$.

Corollary 4.5 (Sample complexity comparison). *Suppose Assumptions 1–5 hold and the algorithm parameters are chosen as in Definition 4.1. Given $n_k = \lceil k^a \rceil$, $a \geq 0$, and $\epsilon > 0$,* ***ReSA*** *requires $O(\epsilon^{-(a+2)/(a+1)})$ SA steps to obtain an ϵ-solution, whereas* ***WaSA*** *requires no more than $C_{\epsilon,1}\epsilon^{-(a+2)/(a+1)} - C_{\epsilon,2}\epsilon^{-\lambda-1/(a+1)}$ SA steps, where $C_{\epsilon,1}$ and $C_{\epsilon,2}$ are such that $C_{\epsilon,1} > C_{\epsilon,2} > 0$.*

Practically, employing these upper bounds to determine when to stop either algorithm is difficult as constants in the expected sub-optimality bounds are unknown. Nevertheless, it is sensible to stop the multi-period scheme when N_k is large enough (e.g. $1/N_k \leq \epsilon$) as the expected sub-optimality bound is controlled to be $O(N_k^{-1})$ in both algorithms.

5 MULTI-PERIOD SA FRAMEWORK WITH BIASED GRADIENT ESTIMATOR

In this section, we consider a more general SO setting; when unbiased G is not available. In Section 5.1, we introduce the SP gradient estimator for the choice of G and analyze the expected sub-optimality and computational costs of **ReSA** and **WaSA** in Section 5.2.

5.1 Simultaneous perturbation gradient estimator

Let $\varepsilon(\mathbf{x},\boldsymbol{\theta},\omega)$ represent the simulation error such that $F(\mathbf{x},\boldsymbol{\theta},\omega) = f(\mathbf{x},\boldsymbol{\theta}) + \varepsilon(\mathbf{x},\boldsymbol{\theta},\omega)$. Clearly, $\mathbb{E}_\omega[\varepsilon(\mathbf{x},\boldsymbol{\theta},\omega)] = 0$ as $f(\mathbf{x},\boldsymbol{\theta}) = \mathbb{E}_\omega[F(\mathbf{x},\boldsymbol{\theta},\omega)]$. The SP estimator of $\nabla_\mathbf{x} f(\mathbf{x}_{k,j},\boldsymbol{\theta}_k)$ requires running simulation replications at two points $\mathbf{x} \pm c_{k,j}\Delta_{k,j}$ given window size $c_{k,j} > 0$ and perturbation vector $\Delta_{k,j} \in \mathbb{R}^d$ sampled from a distribution independently from simulation replications. We consider increasing the number of replications spent to compute the SP gradient estimator as a function of j so that the estimator becomes increasingly more precise as iterations continue. Namely, $s_{k,j}$ i.i.d. replications are run at each of $\mathbf{x} \pm c_{k,j}\Delta_{k,j}$, where $s_{k,j}$ is a non-decreasing sequence on in j. The SP estimator of $\nabla_\mathbf{x} f(\mathbf{x}_{k,j},\boldsymbol{\theta}_k)$ is computed as

$$G(\mathbf{x}_{k,j},\boldsymbol{\theta}_k,\boldsymbol{\omega}_{k,j}) = \left[\frac{\bar{F}^+_{k,j} - \bar{F}^-_{k,j}}{2c_{k,j}(\Delta_{k,j})_1}, \cdots, \frac{\bar{F}^+_{k,j} - \bar{F}^-_{k,j}}{2c_{k,j}(\Delta_{k,j})_d}\right]^\top, \tag{8}$$

where $\boldsymbol{\omega}_{k,j} = \left\{\omega^+_{k,j,1}, \omega^-_{k,j,1}, \omega^+_{k,j,2}, \omega^-_{k,j,2}, \dots, \omega^+_{k,j,s_{k,j}}, \omega^-_{k,j,s_{k,j}}\right\}$ is the collection of random number sequences used to calculate G, $\bar{F}^\pm_{k,j} \triangleq s^{-1}_{k,j}\sum_{h=1}^{s_{k,j}} F(\mathbf{x}_{k,j} \pm c_{k,j}\Delta_{k,j},\boldsymbol{\theta}_k,\omega^\pm_{k,j,h})$, and $(\Delta_{k,j})_l$ denotes the lth element of $\Delta_{k,j}$. Furthermore, we define $f^\pm_{k,j} \triangleq f(\mathbf{x}_{k,j} \pm c_{k,j}\Delta_{k,j},\boldsymbol{\theta}_k)$ and $\bar{\varepsilon}^\pm_{k,j} \triangleq \bar{F}^\pm_{k,j} - f^\pm_{k,j}$. Let $\mathbf{b}_{k,j}(\mathbf{x}_{k,j},\boldsymbol{\theta}_k) \triangleq \mathbb{E}\left[G(\mathbf{x}_{k,j},\boldsymbol{\theta}_k,\boldsymbol{\omega}_{k,j}) - \nabla_\mathbf{x} f(\mathbf{x}_{k,j},\boldsymbol{\theta}_k) \middle| \mathbf{x}_{k,j},\boldsymbol{\theta}_k\right]$ represent the conditional bias of $G(\mathbf{x}_{k,j},\boldsymbol{\theta}_k,\boldsymbol{\omega}_{k,j})$ given $\mathbf{x}_{k,j}$ and $\boldsymbol{\theta}_k$.

Spall [30] presents a set of regularity conditions (see Appendix **??**) under which (8) is strongly consistent as $j \to \infty$ for what we call a single-period problem. In the following, we provide a slightly different set of conditions and assumptions to facilitate the analyses in Sections 5.2.

(C1) $\gamma_{k,j}, c_{k,j} > 0, \forall k,j$; $\gamma_{k,j} \to 0$, $c_{k,j} \to 0$ as $j \to \infty$; $\sum_{j=0}^\infty \gamma_{k,j} = \infty$, $\sum_{j=0}^\infty \left(\frac{\gamma_{k,j}}{c_{k,j}}\right)^2 < \infty$;

(C2) There exist $b_0, b_1, \alpha \in \mathbb{R}$ such that $|(\Delta_{k,j})_l| \le b_0$ a.s., $\mathbb{E}[|(\Delta_{k,j})_l^{-1}|] \le b_1$, and $\mathbb{E}[(\Delta_{k,j})_l^{-2}] \le \alpha$ for all k, j and $l = 1, 2, \dots, d$.

(C3) For each k, j, $\Delta_{k,j} \in \mathbb{R}^d$ is independent of $\{\mathbf{x}_{k,1}, \cdots, \mathbf{x}_{k,j};\ \boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_k\}$ and $(\Delta_{k,j})_l, l = 1, \dots, d$, are i.i.d. and symmetrically distributed around zero. Furthermore, $\{\Delta_{k,1}, \Delta_{k,2}, \cdots, \Delta_{k,j}\}$ are mutually independent.

Variants of **(C1)** and **(C2)** are typically seen in selecting stepsize sequences and perturbation levels (see [19] for instance) to guarantee that the sequence of $\mathbb{E}[\|\mathbf{x}_{k,j} - \mathbf{x}^*_k\|^2|\boldsymbol{\theta}_k]$ converges to zero almost surely as j tends to infinity. The independence condition in **(C3)** makes it easier to analyze the SP gradient estimator's bias and variance. Because **(C1)**–**(C3)** lie within the control of the user, they are referred to as "conditions" instead of assumptions. We make two assumptions in the following to establish bounds on the variance and bias of the SP gradient estimator. In Assumption 7, we use the same notation as in [30] to represent third partial derivatives of f with respect to elements of $\mathbf{x}$; $f^{(3)}(\mathbf{x},\boldsymbol{\theta}) = \partial^3 f(\mathbf{x},\boldsymbol{\theta})/\partial\mathbf{x}^{\otimes 3}$ is an arbitrary third partial derivative and $f^{(3)}_{i_1,i_2,i_3}(\mathbf{x},\boldsymbol{\theta}) \triangleq \partial^3 f(\mathbf{x},\boldsymbol{\theta})/\partial\mathbf{x}^{i_1}\partial\mathbf{x}^{i_2}\partial\mathbf{x}^{i_3}$, where $\mathbf{x}^i$ is the ith element of $\mathbf{x}$.

Assumption 6. *There exists* $\sigma_1 < \infty$ *such that* $\mathrm{Var}[\bar{\varepsilon}^+_{k,j} - \bar{\varepsilon}^-_{k,j}|\mathbf{x}_{k,j},\boldsymbol{\theta}_k,\Delta_{k,j}] \le \sigma_1^2/s_{k,j}$ *a.s. for all* k *and* j.

Assumption 7. *For almost all* $\mathbf{x}_{k,j}$, $f^{(3)}(\mathbf{x},\boldsymbol{\theta})$ *exists. For any* $\boldsymbol{\theta} \in \Theta$, $f^{(3)}(\mathbf{x},\boldsymbol{\theta})$ *is continuous in* $\mathbf{x}$ *for all* $\mathbf{x}$ *in an open neighborhood of* $\mathbf{x}_{k,j}$ *that is not a function of* k, j *or* $\boldsymbol{\omega}_{k,j}$. *Moreover, for any* $\mathbf{x} \in \mathcal{X}$ *and* $\boldsymbol{\theta} \in \Theta$, $|f^{(3)}_{i_1,i_2,i_3}(\mathbf{x},\boldsymbol{\theta})| \le b_2$ *for any* $1 \le i_1, i_2, i_3 \le d$.

The bias of the SP gradient estimator can be reduced to zero by ensuring that $c_{k,j} \to 0$; however, its variance increases as $c_{k,j}$ decreases, but can be reduced by increasing $s_{k,j}$ (see SM Section **??**). Therefore, both $\{c_{k,j}\}$ and $\{s_{k,j}\}$ must be controlled relative to $\{\gamma_{k,j}\}$ to ensure that the resulting $\{\mathbf{x}_k\}$ converges to $\mathbf{x}^*$. In Sections 5.2, we choose the parameter sequences to be in the form of $c_{k,j} = c_0 j^{-\eta}, \eta > 0$, and $s_{k,j} = s_0 j^t, t \geq 0$ for SP gradient estimation in **ReSA**. We relax the integrality requirement for $s_{k,j}$ for expository ease in the remainder of the paper. The choices for η and t for **ReSA** to achieve the optimal convergence rate are discussed under different sets of assumptions. For **WaSA**, both parameter sequences are modified to achieve the same convergence rate.

5.2 Rate Analysis

In this subsection, we analyze the performance of both **ReSA** and **WaSA** when $\mathcal{X}$ is bounded and the SP gradient estimator (8) is adopted for G. Similar to the analyses in Section 4, p (the exponent of the single-period MSE convergence rate) plays an important role in both algorithms when specifying the parameters. But when the SP gradient estimator is adopted, p is additionally constrained by the choices of $c_{k,j}$ and $s_{k,j}$. In the following, Definition 5.1 prescribes the choices of the algorithm and SP gradient estimation parameters for both **ReSA** and **WaSA**.

Definition 5.1. For both **ReSA** and **WaSA**, choose $0 \leq t \leq 1/2$ to control the simulation effort at each SA step and let $p = 2(1+t)/3$, $\gamma_0 > p\mu^{-1}$, $s_0 > 0$, $\tilde{s}_0 > 0$, $c_0 > 0$, and $\tilde{c}_0 > 0$. For **ReSA**, let $s_{k,j} = s_0 j^t$ and $c_{k,j} = c_0 j^{-(1+t)/6}$ for all k and j. For **WaSA**, let $\tilde{\gamma}_0 = 2/\mu, s_{1,j} = s_0 j^t$ for all $j \geq 1$, and $s_{k,j} = \tilde{s}_0(N_{k-1}^\lambda + j - 1)^t$ and $c_{k,j} = \tilde{c}_0(N_{k-1}^\lambda + j - 1)^{-(1+t)/6}$ for all $k \geq 2$ and j.

Observe that p as well as the exponent of j in $c_{k,j}$ are functions of t. When $t = 0$, we obtain $p = 2/3$, which matches the MSE convergence rate of SPSA known in the literature [11]. When $t > 0$, $c_{k,j}$ is driven to zero at a faster rate to reduce the bias in G more aggressively while keeping the variance of G in check by increasing $s_{k,j}$. For any $t \geq 0$, p cannot exceed 1; $t = 1/2$ is the smallest rate of increase for $s_{k,j}$ to obtain $p = 1$, i.e., no incentive to spend larger simulation effort than $t = 1/2$. For a more detailed discussion on the choices of t and p, see SM Section **??**. The following lemma shows that the exponent of the single-period MSE convergence rate is indeed p in this case.

Lemma 5.2 (Single-period MSE convergence rate). *Suppose Assumptions 1, 4, 6, 7 and* **(C1)**–**(C3)** *hold and the algorithm parameters are chosen as in Definition 5.1. Consider $\{\mathbf{x}_k\}$ generated by **ReSA**. Then, the following holds for $k \geq 1$ and $1 \leq j \leq M_k$, $\mathbb{E}[\|\mathbf{x}_{k,j} - \mathbf{x}_k^*\|^2|\boldsymbol{\theta}_k] \leq \max\{\mathbb{E}[\|\mathbf{x}_{k-1} - \mathbf{x}_k^*\|^2|\boldsymbol{\theta}_k], 2^p T, T(\mu\gamma_0 - p)^{-1}\}j^{-p}$ a.s., where T is defined as*

$$T \triangleq \gamma_0^2 d\left(2C_{\mathcal{X}}^2 + 2R^2 c_0^4 + \frac{\alpha\sigma_1^2}{4c_0^2 s_0} + C_{\mathcal{X}}^2 n b_0^2 \alpha\right) + \gamma_0 d R^2 c_0^4 \mu^{-1}. \tag{9}$$

Theorem 5.3 analyzes the performance of **ReSA** when G is a SP gradient estimator. In contrast to Theorem 4.3, here we differentiate the cumulative number of SA steps from the cumulative simulation effort as $t > 0$ is considered.

Theorem 5.3 (**ReSA** with SP gradient estimator). *Suppose Assumptions 1–4, 6, 7 and* **(C1)**–**(C3)** *hold and the algorithm parameters are chosen as in Definition 5.1. Define T as in (9). Consider $\{\mathbf{x}_k\}$ generated by **ReSA**. Then, for all k, $j \geq 1$, the following hold.*

(i) There exists finite $U > 0$ and sequence $\{Q_k'\}$ such that $Q_k' \xrightarrow{k\to\infty} \max\{2^p T, T(\mu\gamma_0 - p)^{-1}\}$ and for any $k \geq 1$,

$$\mathbb{E}\left[f(\mathbf{x}_k, \boldsymbol{\theta}^*) - f(\mathbf{x}^*, \boldsymbol{\theta}^*)\right] \leq L_{\mathcal{X}}\left(UL_\Theta^2/\mu^2 + Q_k'\right)N_k^{-1}.$$

(ii) Given $K > 0$, the cumulative number of SA steps at the end of period K is bounded as $\sum_{k=1}^K M_k \leq \sum_{k=1}^K N_k^{1/p} + K$.

(iii) Given $K > 0$, the cumulative simulation effort at the end of period K is bounded as $\sum_{k=1}^K \sum_{j=1}^{M_k} s_{k,j} \leq \frac{3^{t+1}s_0}{t+1}\sum_{k=1}^K N_k^{3/2}$.

Theorem 5.3 confirms that **ReSA** achieves the optimal convergence rate, $O(N_k^{-1})$, given the parameter choices in Definition 5.1. Regardless of t, the cumulative simulation effort is of the same order. Thus, there is little room to save simulation effort by adjusting t. On the other hand, choosing a larger t reduces the cumulative number of SA steps. Thus, when the projection step is costly, then $t = 1/2$ is preferred. Similar insights can be obtained from the performance analysis for **WaSA** in Theorem 5.4 below.

Theorem 5.4 (**WaSA** with SP gradient estimator). *Suppose Assumptions 1–4, 6, 7 and* **(C1)–(C3)** *hold and the algorithm parameters are chosen as in Definition 5.1. Consider the sequences* $\{\mathbf{x}_k\}$ *and* $\{\boldsymbol{\theta}_k\}$ *generated by* ***WaSA****. Suppose* $T_2 \triangleq \frac{d}{\mu^2}\left(10R^2\tilde{c}_0^4 + 8C_{\mathcal{X}}^2 + 4C_{\mathcal{X}}^2 d b_0^2 \alpha + \frac{\alpha\sigma_1^2}{\tilde{c}_0^2 \tilde{s}_0}\right)$.

(i) *There exists finite* $U > 0$ *and sequence* $\{Q_k'\}$ *such that* $Q_k' \xrightarrow{k\to\infty} T_2$ *and for any* $k \geq 1$,

$$\mathbb{E}\left[f(\mathbf{x}_k, \boldsymbol{\theta}^*) - f(\mathbf{x}^*, \boldsymbol{\theta}^*)\right] \leq L_{\mathcal{X}}\left(L_{\Theta}^2 U/\mu^2 + Q_k'\right) N_k^{-1}.$$

(ii) *Given* $K \geq 2$, *there exists* $0 < \lambda < 1/p$ *sufficiently close to* $1/p$ *such that the cumulative number of SA steps at the end of period* K *is bounded as* $\sum_{k=1}^{K} M_k \leq N_K^{1/p} + 2K - 1$.

(iii) *Given* $K \geq 2$, *there exists* $0 < \lambda < 1/p$ *sufficiently close to* $1/p$ *such that the cumulative simulation effort at the end of period* K *is bounded as*

$$\sum_{k=1}^{K} \sum_{j=1}^{M_k} s_{k,j} \leq \frac{(2^{t+1}+1)\tilde{s}_0}{t+1} N_K^{3/2} + \frac{3^{t+1}s_0 - \tilde{s}_0}{t+1} N_1^{3/2} - \frac{\tilde{s}_0}{t+1} N_2^{3/2} + \frac{(K-1)\tilde{s}_0}{t+1}.$$

Comparing Theorems 5.3 and 5.4, observe that there is a stark difference in the computational cost between **ReSA** and **WaSA**. Notice that the computational saving of the latter is more pronounced with the SP gradient estimator than when an unbiased gradient estimator is available. The following corollary compares the computational complexity for both algorithms to obtain an ϵ-solution.

Corollary 5.5. *Suppose Assumptions 1–4, 6, 7 and* **(C1)–(C3)** *hold and the algorithm parameters are chosen as in Definition 5.1. Given* $n_k = \lceil k^a \rceil$, $a \geq 0$, *and* $\epsilon > 0$, ***ReSA*** *requires* $O(\epsilon^{-1/p-1/(a+1)})$ *SA steps to obtain an* ϵ*-solution, whereas* ***WaSA*** *requires no greater than* $C_{\epsilon,1}\epsilon^{-1/p-1/(a+1)} - C_{\epsilon,2}\epsilon^{-\lambda-1/(a+1)}$ *SA steps, where* $C_{\epsilon,1}$ *and* $C_{\epsilon,2}$ *are such that* $C_{\epsilon,1} > C_{\epsilon,2} > 0$.

6 REGULARIZED RESA SCHEME WITH UNKNOWN STRONG CONVEXITY PARAMETER

In this section, we address the case when the strong convexity parameter, μ, is unknown and extend the **ReSA** framework to a regularized variant, referred to as regularized **ReSA** (**r-ReSA**); notably this scheme does not require utilizing the value of μ in specifying its algorithm parameters. The new scheme relies on the Tikhonov regularization framework [34], which adds a strongly convex function to the original objective function of the problem to induce strong convexity. The resulting regularized problem thus has a known strong convexity parameter. Tikhonov regularization schemes have a long history in the field of optimization theory, assuming relevance when the objective function is merely convex [8]. We define a sequence of regularized versions of $f(\mathbf{x}, \boldsymbol{\theta})$ in $\mathbf{x}$ whose kth element is defined as

$$f_k(\mathbf{x}, \boldsymbol{\theta}) \triangleq f(\mathbf{x}, \boldsymbol{\theta}) + \frac{\mu_k}{2}\|\mathbf{x}\|^2 = \mathbb{E}[F(\mathbf{x}, \boldsymbol{\theta}, \omega)|\boldsymbol{\theta}] + \frac{\mu_k}{2}\|\mathbf{x}\|^2, \tag{10}$$

where $\{\mu_k\}$ is a user-specified positive sequence diminishing to zero. For any $\boldsymbol{\theta}$, $f_k(\mathbf{x}, \boldsymbol{\theta})$ in (10) is $(\mu+\mu_k)$-strongly convex in $\mathbf{x}$; since μ is unknown, we may adopt μ_k as our strong convexity parameter at period k. We first briefly summarize the classical convergence result for the Tikhonov regularization. Consider the sequence of regularized problems at some fixed $\boldsymbol{\theta}$ given by $\{\min_{\mathbf{x}\in\mathcal{X}} f_k(\mathbf{x}, \boldsymbol{\theta})\}$. Because each f_k is strongly convex in $\mathbf{x}$, then $\hat{\mathbf{x}}_k \triangleq \arg\min_{\mathbf{x}\in\mathcal{X}} f_k(\mathbf{x}, \boldsymbol{\theta})$ is the

Algorithm 3 (**r-ReSA**). Regularized re-start multi-period SA

```
Given $\mathbf{x}_0$, $\{\gamma_{0,k}\}$, $p$, $\{N_k\}$, $\{\mu_k\}$; Set $k := 1$;
[1] Compute $\boldsymbol{\theta}_k$ using $N_k$ samples; Set $\mathbf{x}_{k,1} := \mathbf{x}_{k-1}$ and $M_k := \max\left\{1, \left\lceil N_k^{2/p} \right\rceil\right\}$;
[2] $\mathbf{x}_{k,j+1} := \Pi_{\mathcal{X}}\left[\mathbf{x}_{k,j} - \gamma_{k,j}\{G(\mathbf{x}_{k,j}, \boldsymbol{\theta}_k, \omega_{k,j}) + \mu_k \mathbf{x}_{k,j}\}\right]$ given $\gamma_{k,j} = \frac{\gamma_{0,k}}{j}$ for $j = 1, \cdots, M_k$;
[3] $\mathbf{x}_k := \mathbf{x}_{k,M_k+1}$; $k := k + 1$ and go to [1];
```

unique minimizer at k. Moreover, it has been shown that $\hat{\mathbf{x}}_k \xrightarrow{k\to\infty} \hat{\mathbf{x}}$, if $\mu_k \longrightarrow 0$, where $\hat{\mathbf{x}}$ is a least-norm minimizer of $\mathrm{Opt}(\boldsymbol{\theta})$ (cf. [8]); in our problem, $\hat{\mathbf{x}}$ is indeed the unique minimizer of $\mathrm{Opt}(\boldsymbol{\theta})$ due to strong convexity of f.

In the multi-period SA problem, new $\boldsymbol{\theta}_k$ is computed at each period k. Thus, **r-ReSA** tackles the regularized problem

$$\min_{\mathbf{x}\in\mathcal{X}} \left(f(\mathbf{x}, \boldsymbol{\theta}_k) + \frac{\mu_k}{2}\|\mathbf{x}\|^2 \right) \tag{r-Opt$_k(\boldsymbol{\theta}_k)$}$$

for some appropriate choice for $\mu_k > 0$. Akin to **ReSA**, **r-ReSA** solves r-Opt$_k(\boldsymbol{\theta}_k)$ up to the precision determined by N_k by employing M_k SA steps with the known strong convexity parameter, μ_k. Algorithm 3 provides the details of **r-ReSA**. We highlight that in Step [2], the gradient includes the extra term, $\mu_k \mathbf{x}_{k,j}$, arising from regularization. Contrasting with **ReSA**, the stepsize sequence constant, $\gamma_{0,k}$, depends on k in **r-ReSA** as μ_k is updated at each k.

Intuitively, it is sensible to drive $\mu_k \longrightarrow 0$ as k increases so that r-Opt$_k(\boldsymbol{\theta}_k)$ becomes progressively closer to $\mathrm{Opt}(\boldsymbol{\theta}^*)$. Meanwhile, reducing μ_k makes the algorithm perceive r-Opt$_k(\boldsymbol{\theta}_k)$ to be flatter than it actually is; recall that the true strong convexity parameter for r-Opt$_k(\boldsymbol{\theta}_k)$ is $(\mu + \mu_k)$, unbeknownst to the user. Hence, the decreasing sequence, $\{\mu_k\}$, together with a careful choice of $\{M_k\}$, ensures that the sequence of solutions, $\{\mathbf{x}_k\}$, returned by **r-ReSA** at the end of each period, attains the best-possible convergence rate of the expected sub-optimality for $\mathrm{Opt}(\boldsymbol{\theta}^*)$. Definition 6.1 formally states the choices for the algorithm parameters of **r-ReSA**.

Definition 6.1. For **r-ReSA**, let $\mu_k = N_k^{-1/2}$. Depending on the choice for G, set the remaining parameters as follows:

(1) when G is unbiased, let $p = 1$, $\gamma_{0,k} = 1/\mu_k$.
(2) when G is the SP gradient estimator, choose $0 \le t \le 1/2$ and let $s_{k,j} = s_0 j^t$ for some $s_0 > 0$ and $c_{k,j} = c_0 j^{-(1+t)/6}$ for some $c_0 > 0$ for all k and j. Let $p = 2(1+t)/3$ and $\gamma_{0,k} = p_0 \mu_k^{-1}$ for some $p_0 > p$.

The following two theorems state respective convergence results for **r-ReSA** for the two different choices for G.

Theorem 6.2 (**r-ReSA** with unbiased estimator). *Suppose Assumptions 1–5 hold and the algorithm parameters are chosen as in Part (1) of Definition 6.1. Then, the following hold for* $\{\mathbf{x}_k\}$ *generated by* **r-*ReSA***.
(i) For any $k \ge 1$, $\mathbb{E}[f(\mathbf{x}_k, \boldsymbol{\theta}^*) - f(\mathbf{x}^*, \boldsymbol{\theta}^*)] \le O(N_k^{-1})$.
(ii) Given $K > 0$, *the cumulative number of SA steps at the end of period* K *is* $\sum_{k=1}^K M_k = \sum_{k=1}^K N_k^2$.

Theorem 6.3 (**r-ReSA** with SP gradient estimator). *Suppose Assumptions 1–4, 6, 7 and* **(C1)**–**(C3)** *hold and the algorithm parameters are chosen as in Part (b) of Definition 6.1. Then, the following hold for* $\{\mathbf{x}_k\}$ *generated by* **r-*ReSA***.
(i) For any $k \ge 1$, $\mathbb{E}[f(\mathbf{x}_k, \boldsymbol{\theta}^*) - f(\mathbf{x}^*, \boldsymbol{\theta}^*)] \le O(N_k^{-1})$.
(ii) Given $K > 0$, *the cumulative number of SA steps at the end of period* K *is bounded as* $\sum_{k=1}^K M_k = \sum_{k=1}^K N_k^{2/p} + K$.
(iii) Given $K > 0$, *the cumulative simulation effort at the end of period* K *is bounded as* $\sum_{k=1}^K \sum_{j=1}^{M_k} s_{k,j} = \frac{3^{t+1}s_0}{t+1} \sum_{k=1}^K N_k^3$.

When μ_k falls below μ for some k, **r-ReSA** chooses a larger stepsize constant than **ReSA** would have at the same period, which may lead to a larger gradient descent step. This can be seen by contrasting γ_0 in Definitions 4.1 and 5.1 with $\gamma_{0,k}$ in Definition 6.1. When k is large, a smaller value of μ_k may cause the aforementioned gradient-descent steps within the kth period to overshoot the feasible region $\mathcal{X}$ rendering the iterates to be projected back to $\mathcal{X}$. Furthermore, we observe that the computational burden (and therefore the simulation effort) is significantly higher at each step since $M_k = \mathcal{O}(N_k^{2/p})$ (as opposed to $M_k = \mathcal{O}(N_k^{1/p})$ for **ReSA** and **WaSA**). We defer creating a variant of **WaSA** under the regularization scheme for future research.

7 EMPIRICAL PERFORMANCE

In this section, we examine the empirical performances of the proposed algorithms on three examples; in Section 7.1, we apply the algorithms on a set of synthetically constructed strongly convex SO problems that have analytical expressions for the optimal solution of Opt($\boldsymbol{\theta}$) for any $\boldsymbol{\theta}$. To test the algorithms on more realistic simulation settings, we consider a stochastic activity network (SAN) example in Section 7.2 while Section 7.3 revisits the EMS example introduced in Section 1. Note that the SAN problem is strongly convex with known μ, whereas the EMS example is a more general SO problem for which convexity cannot be verified. The numerical results show that **WaSA** outperforms **ReSA** in terms of computational effort, while achieving similar empirical expected sub-optimality. **r-ReSA** achieves the same expected sub-optimality as of **ReSA** and **WaSA**, but requires more computational effort. From the EMS example, we observe that both algorithms perform well even if some of the assumptions cannot be verified for this problem. We also demonstrate robustness of the proposed algorithms when strong convexity fails by applying them to a nonconvex problem in Section 7.4.

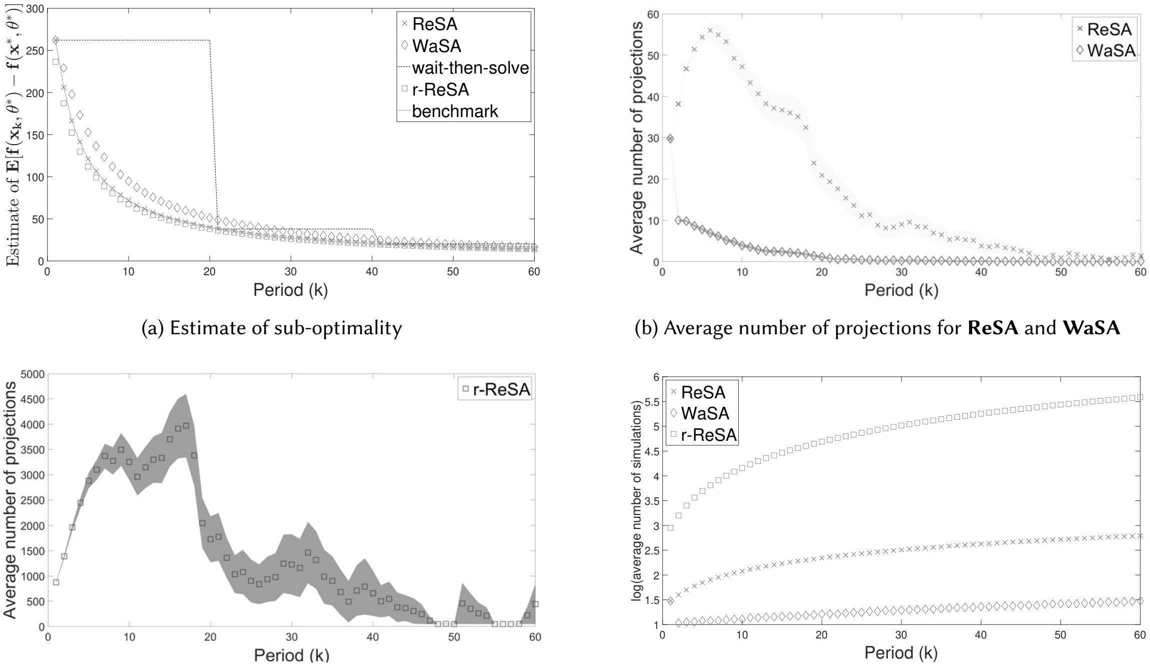
7.1 Stochastic quadratic programming

We apply **ReSA** and **WaSA** under the settings analyzed in Section 4 to a synthetic SO problem that has

$$f(\mathbf{x}, \boldsymbol{\theta}) \;=\; \frac{1}{2}\mathbf{x}^\top V^\top \mathrm{diag}(\mathbf{u}) V\mathbf{x} + \mathbf{x}^\top \mathbf{v}, \tag{11}$$

where $\boldsymbol{\theta} = (\mathbf{u}, \mathbf{v})$, $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d : -5 \le \mathbf{x}_i \le 5, 1 \le i \le d\}$ and V is an $d \times d$ deterministic orthogonal matrix; increasing levels of d from 5 to 100 are tested below. Note that $\mathrm{diag}(\mathbf{u})$ denotes a diagonal matrix whose diagonal entries are given by $\mathbf{u}$. The true parameter vector is denoted by $\boldsymbol{\theta}^* = (\mathbf{u}^*, \mathbf{v}^*)$, where $\mathbf{u}^*$ is the d-dimensional vector whose entries are all equal to 2.5 and $\mathbf{v}^*$ is a deterministic vector whose entries are i.i.d. Uniform$(0, 10)$. Each entry of $\mathbf{u}^*$ represents the mean of an exponential distribution while $\mathbf{v}^*$ is the mean vector of normally distributed input data with a known covariance matrix. The maximum likelihood estimator (MLE) of $(\mathbf{u}^*, \mathbf{v}^*)$ is computed from i.i.d. observations of $Z_{\mathbf{u}} \in \mathbb{R}^d$ and $Z_{\mathbf{v}} \in \mathbb{R}^d$. The entries of $Z_{\mathbf{u}} \in \mathbb{R}^d$ are i.i.d. Exp(rate= 0.4) and $Z_{\mathbf{v}} \sim \mathcal{N}(\mathbf{v}^*, 400 I_d)$, where I_d denotes the $d \times d$ identity matrix. We set $\Theta = \{(\mathbf{u}, \mathbf{v}) : 2 \le \mathbf{u}_i \le 3, -100 \le \mathbf{v}_i \le 100, 1 \le i \le d\}$, therefore, the MLE at the kth period is computed as $\boldsymbol{\theta}_k = (\Pi_\Theta(\bar{Z}_{\mathbf{u}}(k), \bar{Z}_{\mathbf{v}}(k)))$, where $\bar{Z}_{\mathbf{u}}(k)$ and $\bar{Z}_{\mathbf{v}}(k)$ are the corresponding averages of cumulative observations of $Z_{\mathbf{u}}$ and $Z_{\mathbf{v}}$, respectively. In the first period, we assume that 30 observations of $Z_{\mathbf{u}}$ and $Z_{\mathbf{v}}$ are available. For $k \ge 2$, the sample size of the new batch of data, n_k, is generated randomly from discrete uniform$(5, 15)$. Given Θ, this problem has $\mu = 2$. To test the case with an unbiased gradient estimator, stochastic noise $\boldsymbol{\xi} \sim \mathcal{N}(0, I_d)$ is added to the exact gradient, i.e., $G(\mathbf{x}, \boldsymbol{\theta}) = G(\mathbf{x}, (\mathbf{u}, \mathbf{v})) = V^\top \mathrm{diag}(\mathbf{u}) V\mathbf{x} + \mathbf{v} + \boldsymbol{\xi}$.

We test **ReSA** and **WaSA** with $\gamma_0 = \tilde{\gamma}_0 = 0.5$ and $\lambda = 0.995$, while the rest of the algorithm parameters are chosen as in Definition 4.1. We compare the algorithms with **r-ReSA** and a newly defined scheme referred to as “wait-then-solve,” which updates $\boldsymbol{\theta}_k$ less frequently and implements SA only when $\boldsymbol{\theta}_k$ has been updated; during the remaining periods, the scheme “waits” while the decision $\mathbf{x}_k$ stays unchanged. The wait-then-solve scheme is designed to show the increase in regret when we do not adapt each period’s decision to $\{\boldsymbol{\theta}_k\}$ and only do so intermittently.



(a) Estimate of sub-optimality

(b) Average number of projections for **ReSA** and **WaSA**

(c) Average number of projections for **r-ReSA**

(d) Logarithmic (base 10) average number of simulations

Fig. 2. Comparison of **ReSA**, **WaSA** and **r-ReSA** on a 100-dimensional problem with 200 macro runs

Figure 2 compares the performances of **ReSA**, **WaSA**, **r-ReSA** and wait-then-solve applied to (11) with $d = 100$ from 200 macro runs. In each macro run, we generate $\mathbf{x}_0$ uniformly from $\mathcal{X}$. Figure 2a shows the trajectory of each period's estimated expected sub-optimality of the four algorithms taking the average of $f(\mathbf{x}_k, \boldsymbol{\theta}^*) - f(\mathbf{x}^*, \boldsymbol{\theta}^*)$ obtained from 200 macro runs. Thus, the area under the curve represents the estimated cumulative regret for each algorithm. The benchmark represents $\mathbb{E}[f(\mathbf{x}_k^*, \boldsymbol{\theta}^*) - f(\mathbf{x}^*, \boldsymbol{\theta}^*)]$; for this problem, $\mathbf{x}_k^*$ can be computed exactly given $\boldsymbol{\theta}_k$. Notice that the expected sub-optimality converges slightly faster for **ReSA** than **WaSA** in earlier periods; however, **WaSA** catches up after $k = 40$. The estimated expected sub-optimality of **ReSA** coincides with the benchmark, but that of **r-ReSA** differs from the benchmark. The distinction of **r-ReSA** from the benchmark arises because it solves r-Opt$_k(\boldsymbol{\theta}_k)$ rather than Opt$_k(\boldsymbol{\theta}_k)$. Although in this example, **r-ReSA** appears to achieve smaller expected sub-optimality, however, this is by coincidence and is not a general trend; we discuss this more in detail in Section 7.2. The wait-then-solve scheme updates $\boldsymbol{\theta}_k$ at $k = 1$, 21 and 41; its estimated expected sub-optimality is significantly larger than other algorithms' when $\boldsymbol{\theta}_k$ is not updated for a long time leading to larger cumulative regret. We have also tested a version of "wait-then-solve" that expends the cumulative number of SA steps taken by **ReSA** since its last update and this scheme performs identically to the version presented in Figure 2a. This demonstrates that there is little gain in expending more simulation effort in reducing the expected sub-optimality without updating $\boldsymbol{\theta}_k$.

Figures 2b and 2c show the average number of Euclidean projections **ReSA**, **WaSA**, and **r-ReSA** takes in each period. The shaded area around each line shows the two standard-error band on the average projections at each period calculated from 200 macro runs. The number of projections **ReSA** employs is four times that of **WaSA** in periods

Table 2. Estimated expected sub-optimality under different d at period $k = 100$ computed from 200 macro runs. For **WaSA**, $\lambda = 0.995$ is used; standard errors are presented in parentheses.

	$\mathbb{E}[f(\mathbf{x}_k, \boldsymbol{\theta}^*) - f(\mathbf{x}^*, \boldsymbol{\theta}^*)]$				Cumulative number of simulations ($\times 10^4$)
	$d = 5$	$d = 10$	$d = 50$	$d = 100$	
ReSA	0.44 (0.02)	0.89 (0.03)	4.20 (0.06)	8.72 (0.09)	5.25 (0.01)
WaSA	0.47 (0.02)	0.99 (0.03)	4.62 (0.07)	9.58 (0.10)	**0**.27 (0.00)
r-ReSA	0.43 (0.02)	0.88 (0.03)	4.12 (0.06)	8.58 (0.09)	3593.51 (16.86)
$\mathbb{E}[f(\mathbf{x}_k^*, \boldsymbol{\theta}^*) - f(\mathbf{x}^*, \boldsymbol{\theta}^*)]$	0.44 (0.02)	0.89 (0.03)	4.19 (0.06)	8.70 (0.09)	

$2 \leq k \leq 20$, while both decrease to zero in the later periods. Notice that the y-axis scales are significantly different in Figures 2b and 2c indicating that **r-ReSA** takes orders of magnitude larger number of projections. In Section 6, we discuss that this behavior may be anticipated for **r-ReSA** when μ_k falls below the actual μ. Indeed, in this example, μ_k at all k is smaller than μ. Figure 2d displays the logarithmic (base 10) average number of simulations of the three algorithms required for gradient estimation. Clearly, **WaSA** saves more simulation effort than the other two algorithms. Although in this example, $\mathcal{X}$ is a simple hyperbox, the projection operation may be costly when $\mathcal{X}$ is complex. Therefore, savings in both the number of projections and simulations are critical when either operation is computationally expensive. In such a case, **WaSA** has a significant advantage over both **ReSA** and **r-ReSA**. Although not requiring the value of μ is a clear advantage of **r-ReSA**, this example demonstrates that it comes at a significant computational cost.

Table 2 demonstrates robustness of our algorithms for varying problem dimensions. For $d = 5, 10, 50$, and 100, we test all four algorithms under the same sequence of streaming data. All statistics are collected at period $k = 100$ and averaged over 200 macro runs with their standard errors presented in parentheses. The last row of Table 2 shows the estimated expected sub-optimality of $\mathbf{x}_k^*$, which serves as a benchmark. For all d, the expected sub-optimality of **ReSA** is statistically indistinguishable from the benchmark, while **WaSA** performs slightly worse than the benchmark for $d = 50$ and 100. Nevertheless, the difference in the expected sub-optimality between **WaSA** and **ReSA** is dominated by the magnitude of $\mathbb{E}[f(\mathbf{x}_k^*, \boldsymbol{\theta}^*) - f(\mathbf{x}^*, \boldsymbol{\theta}^*)]$. We point out that the number of simulations expended by each algorithm depends on $\{N_k\}$ and choices of algorithm parameters, but not on d. Since n_k is sampled from the same discrete uniform distribution for all d, we simply present the cumulative number of simulations employed by each algorithm by the 100th period averaged across all d in the last column of Table 2. The computational effort of the projections scales up linearly, d. However, as the feasible region is a hyperbox in this example, the projection effort is negligible and thus is not reported here. Notice that **WaSA** takes less than 5% of the projections required by **ReSA**. As observed in Figure 2d, **r-ReSA** requires orders of magnitude more simulation effort.

Lastly, we examine sensitivity of **WaSA**'s performance to the choice of λ. Table 3 confirms that there is little difference in the expected sub-optimality for different choices of λ, while the cumulative number of simulations can be significantly reduced by choosing λ close to 1.

7.2 Stochastic Activity Network

In this section, we consider a SAN problem, which originally appeared in [2] and is archived as a test problem at SimOpt [20]. We refer the readers to [13] for the structure of the network, which contains 13 arcs and 9 vertices. Each arc is associated with an activity whose operating time is exponentially distributed random variable. The total completion time of the activity network is equivalent to the longest path from the source (node a) to the sink (node i). As discussed

Table 3. Estimated expected sub-optimality **WaSA** achieves with varying λ values at the 50th period computed from 30 macro runs; standard errors presented in parentheses.

	$d = 5$	$d = 10$	$d = 50$	$d = 100$	Cumulative number of simulations ($\times 10^4$)
$\lambda = 0.5$	0.34 (0.03)	0.64 (0.05)	3.51 (0.13)	7.09 (0.20)	3.19 (0.01)
$\lambda = 0.75$	0.34 (0.04)	0.64 (0.05)	3.52 (0.13)	7.10 (0.21)	2.70 (0.01)
$\lambda = 0.9$	0.34 (0.04)	0.64 (0.05)	3.53 (0.13)	7.08 (0.20)	1.67 (0.01)
$\lambda = 0.95$	0.35 (0.04)	0.65 (0.05)	3.58 (0.14)	7.15 (0.20)	1.03 (0.00)
$\mathbb{E}[f(\mathbf{x}_k^*, \boldsymbol{\theta}^*) - f(\mathbf{x}^*, \boldsymbol{\theta}^*)]$	0.34 (0.04)	0.64 (0.05)	3.51 (0.13)	7.08 (0.20)	

in [2], one can derive an IPA gradient estimator for the completion time with respect to the parameters of the activity times. We utilized the codes provided by [20] to compute the IPA gradient estimator in our experiments.

We consider the setting where the mean parameters of the first six activity times are our decision variables, i.e., $\mathbf{x} = (x_1, x_2, \ldots, x_6)^\top$, where the cost of each x_i is $1/x_i$. The means of the remaining seven activity times are to be estimated from streaming data via maximum likelihood estimation, i.e., $\boldsymbol{\theta} \in \mathbb{R}^7$. We denote the simulated total completion time of the network given $\mathbf{x}$ and $\boldsymbol{\theta}$ by $T(\mathbf{x}, \boldsymbol{\theta}, \omega)$. The following objective function balances the expected total completion time and the cost:

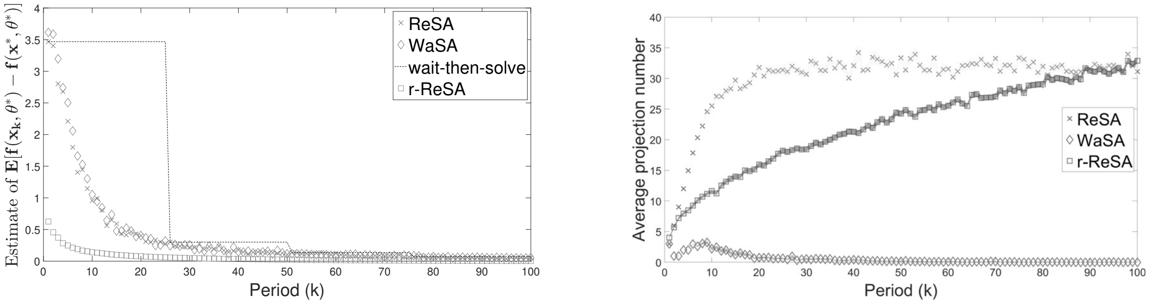
$$f(\mathbf{x}, \boldsymbol{\theta}) \; = \; \mathbb{E}[T(\mathbf{x}, \boldsymbol{\theta}, \omega) | \boldsymbol{\theta}] + \sum\nolimits_{i=1}^{6} \frac{1}{x_i}, \tag{12}$$

where $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^6 | 0.5 \leq x_i \leq 3, \text{ for } i = 1, \ldots, 6\}$. Notice that given $\boldsymbol{\theta}$, smaller x_i makes $\mathbb{E}[T(\mathbf{x}, \boldsymbol{\theta}, \omega)|\boldsymbol{\theta}]$ smaller but increases the cost function, $\sum_{i=1}^{6} 1/x_i$. The expectation $\mathbb{E}[T(\mathbf{x}, \boldsymbol{\theta})]$ is convex in $\mathbf{x}$ [13] while the cost function, $\sum_{i=1}^{6} 1/x_i$, is strongly convex with with parameter $\mu = \frac{2}{27}$.

In the following, we test **ReSA**, **WaSA**, and **r-ReSA** with $\gamma_0 = \tilde{\gamma}_0 = 13.5$ and $\lambda = 0.95$. All entries of $\boldsymbol{\theta}^* \in \mathbb{R}^7$ are equal to one and $n_k = 3$ for all $1 \leq k \leq 100$. Within each macro run, the initial guess $\mathbf{x}_0$ is uniformly generated from $\mathcal{X}$. The optimal function value $\mathbb{E}[f(\mathbf{x}^*, \boldsymbol{\theta}^*)]$ is estimated by implementing SA on Opt$(\boldsymbol{\theta}^*)$ with 1000 steps.

Figure 3a shows the estimated expected sub-optimality over 50 macro runs; **ReSA** and **WaSA** perform similarly for all $1 \leq k \leq 100$. Here, the wait-then-solve scheme updates $\boldsymbol{\theta}_k$ and optimizes at $k = 1, 26, 51$ and 76, which shows larger cumulative regret as seen in Section 7.1 as well. In earlier periods, **r-ReSA** outperforms the others in average sub-optimality, however, the difference becomes less apparent in later periods. Again, we emphasize that **r-ReSA** is targeting the regularized problem r-Opt$_k(\boldsymbol{\theta}_k)$. Since this is a minimization problem, the regularization term, $\frac{\mu_k \|\mathbf{x}\|^2}{2}$, tends to force $\mathbf{x}$ to have smaller entries, which appears to benefit the average sub-optimality for the particular objective function in (12). Figure 3b reveals that in this problem **WaSA** has fewer than 5 projections at the first 20 periods and almost zero for the rest; while **ReSA** has a larger number of projections than **r-ReSA** while the latter catches up in the later periods. This may be misleading at first glance since the number of SA steps that **r-ReSA** takes is a roughly square of the number used in **ReSA** at each period. Since we have relatively large $\gamma_0 = 13.5$ and the **ReSA** stepsize is γ_0/j for $j = 1, 2, \cdots$ at each period, the first few gradient descent steps tend to land $\mathbf{x}_{k,j}$ outside the feasible region, $\mathcal{X}$, thus requiring a subsequent projection. On the other hand, **r-ReSA** takes $\gamma_{0,k}/j$ as the stepsize. Recall that $\gamma_{0,k} = \mu_k^{-1} = \sqrt{N_k}$ is small at the beginning and thus leads to fewer projections; when k increases, $\gamma_{0,k} = \sqrt{N_k}$ grows accordingly, resulting in more projections. For a similar reason, **WaSA** takes fewer projections as its stepsize, defined as $\gamma_0/(N_{k-1}^{\lambda} + j - 1)$, cannot grow significantly with N_{k-1} in the denominator. The shaded area around each line shows the two standard-error band on the average projections at each period calculated from 50 macro runs.

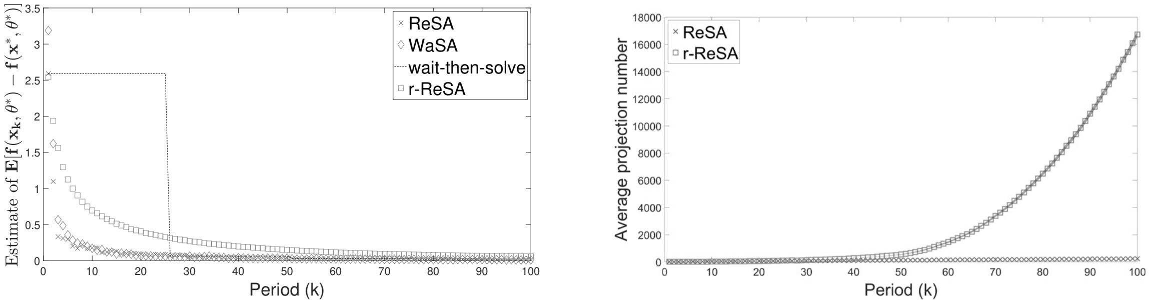
To examine the sensitivity of **r-ReSA** to the structure of the objective function, we slightly modify the cost function in (12) to obtain the new objective function: $f(\mathbf{x}, \boldsymbol{\theta}) = \mathbb{E}[T(\mathbf{x}, \boldsymbol{\theta}, \omega)|\boldsymbol{\theta}] + \sum_{i=1}^{6} \frac{5}{x_i}$, whose strong convexity parameter is



(a) Estimate of sub-optimality

(b) Average number of projections

Fig. 3. Performance of the algorithms applied to the SAN problem averaged over 50 macro runs



(a) Estimate of sub-optimality

(b) Average number of projections

Fig. 4. Performance of the algorithms applied to the modified SAN problem (larger μ) averaged over 50 macro runs

$\mu = \frac{10}{27}$, five times that of (12). We choose $\gamma_0 = \tilde{\gamma}_0 = 2.7$ accordingly. Under the same sequence of streaming data as in Figure 3, the four algorithms' performances using the new objective function are compared in Figure 4. In Figure 4a, the average sub-optimality of **WaSA** and **ReSA** are smaller compared to Figure 3a, which matches our theoretical results. On the other hand, **r-ReSA** is significantly outperformed by **ReSA** and **WaSA** in earlier periods. With the scaled-up cost function, the new objective function is more sensitive to smaller entries of **x** while the regularization term in r-Opt$_k(\boldsymbol{\theta}_k)$ indeed works against **r-ReSA** in this case. In sharp distinction with Figure 3b, the average number of projections in **r-ReSA** is far greater than that of **ReSA** in Figure 4b as the stepsize γ_0/j is significantly smaller than $\gamma_{0,k}/j = \sqrt{N_k}/j$ for larger k with the new μ. **WaSA** is excluded from Figure 4b as it performs significantly fewer projections than the other two algorithms.

7.3 Emergency Medical Service Mobile Station Location Problem

In this section, we examine the performance of **ReSA** and **WaSA** on a realistic simulation optimization application featuring a regional EMS mobile station location problem, a simplified version of a case study conducted in Centre County, PA [38]. The purpose of this example is to demonstrate the robustness of our algorithms even when the mathematical properties of $f(\mathbf{x}, \boldsymbol{\theta})$ are unknown so that the assumptions cannot be verified. The versions of **ReSA** and **WaSA** with the SP gradient estimator examined in Section 5 are applied to this problem.

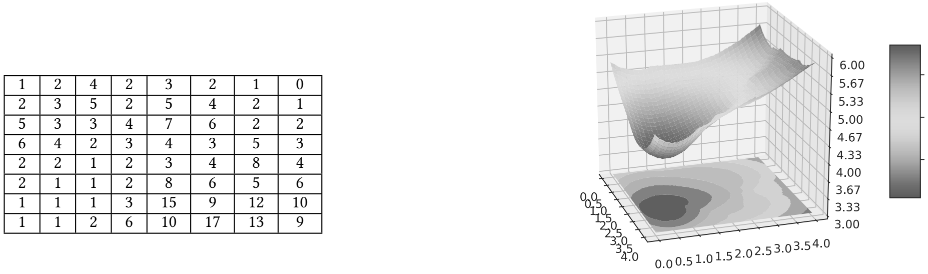
The objective of the problem is to determine the location, $\mathbf{x}$, of a single mobile dispatching station in the county that minimizes the average response time (ART) as described in Section 1. We simplify the solution space to a 2-dimensional box, $\mathcal{X} = [0, 4]^2$, in lieu of the Centre County map. In this version, the hospital is located at $(3.75, 3.75)$. In addition to the mobile station, there are two permanent dispatching stations located at $(2.25, 3.25)$ and $(3.25, 1.75)$, respectively. There are two types of emergency calls and the corresponding types of ambulances. The first type is Advanced Life Support (ALS), which requires more advanced equipment to support patients in critical conditions. The second type is Basic Life Support (BLS). An ALS ambulance may serve both ALS and BLS patients, however, a BLS ambulance may only serve BLS patients. Each station has one ambulance of each type. Once a call is received, an ambulance is dispatched from the nearest station with availability to the patient's location and perform the first aid upon arrival. For BLS calls, we first check if a BLS ambulance is available at the nearest station. If not, an ALS ambulance is dispatched, if available. Otherwise, the system checks the next nearest station's availability. When there is no available ambulance in any of the three stations, the patient joins an ALS or BLS (virtual) service queue. Depending on the severity of the case, the patient may or may not be transferred to the emergency room (ER). When the ambulance is freed, it is dispatched to a patient's location if the service queue is nonempty, or return to its original station location.

A discrete-event simulator is implemented to estimate the ART given a mobile station's location, where a single replication spans five weeks. We assume the following input distributions are known: (i) 30% of the patients are ALS-type; (ii) the-first aid time and transfer time at the ER are exponentially distributed with means 10 and 5 minutes, respectively; and (iii) the travel time between two locations is distributed as Erlang with 6 phases and its mean is 2.7 minutes times their Manhattan distance. We assume that the arrival process of the emergency calls is a spatio-temporal Poisson point process, where the hourly arrival rates on the weekdays and weekends are known as 1.9 and 4.5, respectively, however, the spatial distribution of the emergency calls is unknown and therefore needs to be learned. Furthermore, the map is divided into a 8×8 grid and within each box on the grid, an incoming call's location is assumed to be uniformly distributed. Hence, the estimation of the arrival process boils down to finding the splitting probability vector $\boldsymbol{\theta}^* \in \mathbb{R}^{64}$, which determines the box each incoming emergency call belongs to. The "true" relative frequency of emergency calls from all 64 boxes are given in Figure 5a; normalizing the frequencies gives $\boldsymbol{\theta}^*$.

At the beginning of each period, the emergency call data are generated from the Poisson point process described above; we treat them as streaming data collected from the system to demonstrate our multi-period SA framework. In the kth period, the maximum likelihood estimator, $\boldsymbol{\theta}_k$, of $\boldsymbol{\theta}^*$ is calculated from the cumulative observations, which is simply an empirical probability estimate computed from observed frequencies. The streaming data size in each period ranges from 100 to 150 with the initial sample size $n_1 = 500$. We assume that the location of the mobile station can be updated at each period.

Unlike the SAN example in Section 7.2, it is not guaranteed that the objective function is strongly convex for the EMS example. To observe the function, we discretized the map into a 100×100 grid and evaluated the ART under the true probability distributions assuming the mobile station is located at each vertex of the grid via Monte Carlo simulation. The Monte Carlo simulation budget was chosen such that the best and the second-best vertices can be distinguished with 95% confidence. Figure 5b shows the estimated $f(\mathbf{x}, \boldsymbol{\theta}^*)$ surface, which appears to be (at least locally near the global optimum) strongly convex.

For both **ReSA** and **WaSA**, we chose $t = 0$ and the rest of the algorithm parameters were chosen as in Definition 5.1. We set $s_0 = \tilde{s}_0 = 200$ to reduce the variance of the gradient estimator, however, together with the large streaming data size



1	2	4	2	3	2	1	0
2	3	5	2	5	4	2	1
5	3	3	4	7	6	2	2
6	4	2	3	4	3	5	3
2	2	1	2	3	4	8	4
2	1	1	2	8	6	5	6
1	1	1	3	15	9	12	10
1	1	2	6	10	17	13	9

(a) Relative frequency of emergency calls

(b) Mont Carlo estimates of the ART given the mobile station's location from 500 replications

Fig. 5. Problem characteristics of the EMS example.

Table 4. Monte Carlo estimates of $\mathbb{E}[f(\mathbf{x}_k, \boldsymbol{\theta}^*)]$ and cumulative number of SA steps. Standard errors are presented in parentheses.

Period	**ReSA**		**WaSA**	
	$\mathbb{E}[f(\mathbf{x}_k, \boldsymbol{\theta}^*)]$	Cumulative number of SA steps	$\mathbb{E}[f(\mathbf{x}_k, \boldsymbol{\theta}^*)]$	Cumulative number of SA steps
$k = 25$	4.0517(0.0017)	226.47(1.23)	4.2105(0.0243)	**44.23**(0.26)
$k = 50$	4.0477(0.0008)	1061.27(4.71)	4.0507(0.0007)	**173.07**(0.94)
$k = 75$	4.0489(0.0008)	2737.90(11.63)	4.0470(0.0009)	**435.13**(2.09)
$k = 100$	4.0470(0.0008)	5443.17(21.14)	4.0473(0.0007)	**873.97**(3.76)

N_k, this makes M_k large. To reduce the computational burden, we adopt scaling factor S when determining M_k. For **ReSA**, we choose $M_k = \lceil (N_k/S)^{1/p} \rceil$ with other parameters unchanged. For **WaSA**, we choose $M_k = \lceil (N_k/S)^{1/p} - (N_{k-1}/S)^{\lambda} \rceil$ and adjust the stepsizes and window sizes as $\gamma_{k,j} = 2\mu^{-1}\left((N_{k-1}/S)^{\lambda} + j - 1\right)^{-1}$ and $c_{k,j} = \left((N_{k-1}/S)^{\lambda} + j - 1\right)^{-p/4}$. Note that S does not affect the convergence rate, but affects the constant in the upper bound for the expected sub-optimality in Theorems 5.3 and 5.4. For $t = 0$, the feasible range for λ is $0 < \lambda < 1.5$; we employed $\lambda = 1.45$ in our experiments below. Recall that Definition 5.1 requires $\gamma_0 > p\mu^{-1}$ and $\tilde{\gamma}_0 = 2\mu^{-1}$ but μ is unknown in this setting. Thus, we set $\gamma_0 = \tilde{\gamma}_0 = 10$ as a conservative choice.

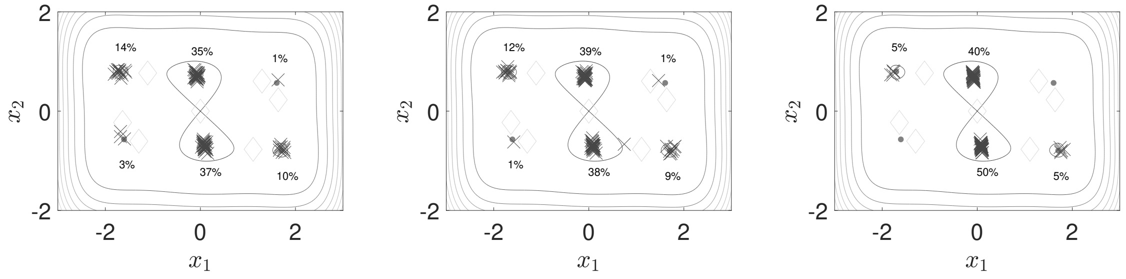
Table 4 presents the estimated $\mathbb{E}[f(\mathbf{x}_k, \boldsymbol{\theta}^*)]$ as well as the cumulative number of SA steps after $k = 25, 50, 75$, and 100 iterations for both algorithms. Note that $\mathbb{E}[f(\mathbf{x}^*, \boldsymbol{\theta}^*)]$ is estimated to be 4.04 from the Monte Carlo simulation shown in Figure 5b. We observe that although at the beginning **ReSA** outperforms **WaSA** in solution quality, **WaSA** catches up in the end. Considering the computational savings, **WaSA** is significantly superior, consistent with our analysis in Section 5. Unlike in Section 7.1, a single replication of the discrete-event simulator for the EMS problem is far more time-consuming; thus, the computational benefit of **WaSA** is even more pronounced here.

7.4 A nonconvex function: six-hump camel function

In this section, we consider an instance of a nonconvex function, the six-hump Camel function [31]:

$$f(\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{3}x_1^6 - 2.1x_1^4 + \theta_1 x_1^2 + x_1 x_2 + \theta_2 x_2^4 + \theta_3 x_2^2 \tag{13}$$

defined on $\mathcal{X} = \{\mathbf{x} = (x_1, x_2) | x_1 \in [-3, 3], x_2 \in [-2, 2]\}$. Following [5], we set $\boldsymbol{\theta}^* = (4, 4, -4)^\top$. We assume $\boldsymbol{\theta}_k$ is estimated via MLE computed from i.i.d. observations of $\mathcal{N}(\boldsymbol{\theta}^*, \mathrm{diag}((20^2, 25^2, 30^2)))$. Within $\mathcal{X}$, $f(\bullet, \boldsymbol{\theta}^*)$ is locally strongly convex at its two global minimizers, $(0.0898, -0.7126)$ and $(-0.0898, 0.7126)$, and its four local minimizers,



(a) **ReSA** (b) **WaSA** (c) **r-ReSA**

Fig. 6. Scatter plot of $\mathbf{x}_{800}$ obtained from 100 macro runs of **ReSA**, **WaSA** and **r-ReSA** on the Six-Hump Camel function when $\sigma = 2$.

$(-1.7035, 0.7960)$, $(1.7035, -0.7960)$, $(1.6071, 0.5686)$ and $(-1.6071, -0.5686)$. In addition, $f(\bullet, \boldsymbol{\theta}^*)$ has seven saddle points; see Figure 6. Given $\mathbf{x}$, we assume that the unbiased gradient estimator, $G(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\xi}) = \nabla_{\mathbf{x}} f(\mathbf{x}, \boldsymbol{\theta}) + \boldsymbol{\xi}$, is available, where each entry of $\boldsymbol{\xi}$ follows $\mathcal{N}(0, \sigma^2(2 + f(\mathbf{x}, \boldsymbol{\theta}))^2)$; notice the dependence of the variance on $f(\mathbf{x}, \boldsymbol{\theta})$.

We tested **ReSA**, **WaSA**, and **r-ReSA** with $\gamma_0 = \tilde{\gamma}_0 = \frac{1}{4}$ and $\lambda = 0.995$. The incoming data size of each period, represented by n_k, is geometrically distributed with success probability $\frac{1}{4}$. In each macro run, $\mathbf{x}_0$ is sampled uniformly in $\mathcal{X}$. Figure 6a, 6b and 6c show the scatter plots of the solutions returned from 100 macro runs of the three algorithms after 800 periods when $\sigma = 2$. All local and global minimizers are marked with circles (two global minimizers are in the middle) while the saddle points are marked as diamonds. The percentage near each minimizer shows the proportion of number of solutions that lie in the corresponding convex valley. Observe that except for one macro run for **WaSA**, all macro runs converge to the vicinity of global and local optima as prescribed by the theory. While 90% of the macro-runs generated by **r-ReSA** converge to the global optimum, a non-negligible fraction of macro runs end up in local minima for **ReSA** and **WaSA**. As seen in the SAN example with the objective function (12), **r-ReSA** benefits from the regularization term as the global optima have smaller norms than local optima.

8 CONCLUDING REMARKS AND FUTURE WORK

In this paper, we consider a multi-period SO problem where simulation model parameters are estimated with increasing precision as more input data accumulate over the decision periods. Focusing on SO problems defined on the continuous feasible solution space, we propose two multi-period SA schemes: **ReSA** and **WaSA**. The key distinction between the two algorithms lies in the choice of the stepsize sequence and the number of SA steps employed in each period; **ReSA** restarts the stepsize sequence at every period while **WaSA** calibrates the stepsize sequences across all periods as a function of the streaming data size leading to a significantly fewer number of SA steps for later periods compared to **ReSA**. Under a suitable strong convexity requirement on f, both **ReSA** and **WaSA** achieve the best-possible convergence rate in the expected sub-optimality when either an unbiased gradient estimator or the SP gradient estimator is employed. Additionally, the bounds on computational effort derived for **WaSA** grow far slower as opposed to their **ReSA** counterparts. This benefit becomes more pronounced when the SP gradient estimator is employed. In addition, we present a regularized **r-ReSA** variant that does not necessitate knowing the strong convexity parameter. We show that under suitable choices of the regularization sequence and the number of SA steps, the resulting expected sub-optimality error diminishes at the best-possible rate. Experiment results support these analyses; in particular, on the EMS example, **WaSA** consumed less than a sixth of the simulation effort taken by **ReSA** while producing solutions of similar estimated sub-optimality.

This work represents a first step towards investigating a broad range of multi-period SO problems under more general settings. An important question lies in extending these techniques to contend with non-parametric input modeling. In addition, we intend to consider settings where the input-generating processes are afflicted by non-stationarity. Extending the **WaSA** scheme to the regularized problem will be explored as well. Finally, we intend to examine how we may contend with relaxing assumptions in the problem class such as convexity and smoothness.

ACKNOWLEDGMENTS

He and Song are supported by National Science Foundation grants DMS-1854659 and CAREER CMMI-2045400. Shanbhag gratefully acknowledges support from ONR Grant N00014-22- 1-2589 and DOE Grant DE-SC0023303.

REFERENCES

[1] Hesam Ahmadi and Uday V Shanbhag. 2020. On the resolution of misspecified convex optimization and monotone variational inequality problems. *Computational Optimization and Applications* 77, 1 (2020), 125–161.

[2] Athanassios N Avramidis and James R Wilson. 1996. Integrated variance reduction strategies for simulation. *Operations Research* 44, 2 (1996), 327–346.

[3] Gian Italo Bischi, Ahmad K Naimzada, and Lucia Sbragia. 2007. Oligopoly games with local monopolistic approximation. *Journal of economic behavior & organization* 62, 3 (2007), 371–388.

[4] Gian Italo Bischi, Lucia Sbragia, and Ferenc Szidarovszky. 2008. Learning the demand function in a repeated Cournot oligopoly game. *Int. J. Systems Science* 39, 4 (2008), 403–419.

[5] Franklin H Branin. 1972. Widely convergent method for finding multiple solutions of simultaneous nonlinear equations. *IBM Journal of Research and Development* 16, 5 (1972), 504–522.

[6] Canan G. Corlu and Bahar Biller. 2013. A subset selection procedure under input parameter uncertainty. In *Proceedings of the 2013 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, 463–473.

[7] Canan G. Corlu and Bahar Biller. 2015. Subset Selection for Simulations Accounting for Input Uncertainty. In *Proceedings of the 2015 Winter Simulation Conference* (Huntington Beach, CA). IEEE Press, Piscataway, NJ, 437–446.

[8] Francisco Facchinei and Jong-Shi. Pang. 2003. *Finite Dimensional Variational Inequalities and Complementarity Problems: Vols I and II.* Springer-Verlag, NY, Inc.

[9] Weiwei Fan, L. Jeff Hong, and Xiaowei Zhang. 2020. Distributionally Robust Selection of the Best. *Management Science* 66, 1 (2020), 190–208.

[10] Michael C. Fu. 2006. Chapter 19 Gradient Estimation. In *Simulation*, Shane G. Henderson and Barry L. Nelson (Eds.). Handbooks in Operations Research and Management Science, Vol. 13. Elsevier, 575–616.

[11] Michael C Fu. 2015. *Handbook of simulation optimization*. Vol. 216. Springer.

[12] Siyang Gao, Hui Xiao, Enlu Zhou, and Weiwei Chen. 2017. Robust ranking and selection with optimal computing budget allocation. *Automatica* 81 (2017), 30–36.

[13] Shane G. Henderson. 2013. Stochastic Activity Network (SAN) Duration. https://github.com/simopt-admin/simopt/blob/master/MATLAB/Problems/SAN/SAN_Duration.pdf.

[14] Dmitry Ivanov and Alexandre Dolgui. 2020. A digital supply chain twin for managing the disruption risks and resilience in the era of Industry 4.0. *Production Planning & Control* (2020), 1–14.

[15] Hao Jiang and Uday V. Shanbhag. 2013. On the solution of stochastic optimization problems in imperfect information regimes. In *Proceedings of the 2013 Winter Simulations Conference*. IEEE Press, Piscataway, NJ, 821–832.

[16] Hao Jiang and Uday V. Shanbhag. 2015. Data-driven schemes for resolving misspecified MDPs: asymptotics and error analysis. In *Proceedings of the 2015 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, 3801–3812.

[17] Hao Jiang and Uday V. Shanbhag. 2016. On the Solution of Stochastic Optimization and Variational Problems in Imperfect Information Regimes. *SIAM Journal on Optimization* 26, 4 (2016), 2394–2429.

[18] Hao Jiang, Uday V. Shanbhag, and Sean P. Meyn. 2018. Distributed Computation of Equilibria in Misspecified Convex Stochastic Nash Games. *IEEE Trans. Autom. Control.* 63, 2 (2018), 360–371.

[19] Jack Kiefer and Jacob Wolfowitz. 1952. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics* 23, 3 (1952), 462–466.

[20] SimOpt Library. 2019. Simulation Optimization (SimOpt) Library. http://github.com/simopt-admin/simopt/wiki.

[21] Tianyi Liu, Yifan Lin, and Enlu Zhou. 2021. A Bayesian Approach to Online Simulation Optimization with Streaming Input Data. In *Proceedings of the 2021 Winter Simulation Conference*. 1–12.

[22] Matthew Mikell and Jen Clark. 2018. *Cheat sheet: What is Digital Twin?* Technical Report. IBM.

[23] Raghu Pasupathy. 2010. On Choosing Parameters in Retrospective-Approximation Algorithms for Stochastic Root Finding and Simulation Optimization. *Operations Research* 58, 4,1 (2010), 889–901.

[24] Michael Pearce and Juergen Branke. 2017. Efficient expected improvement estimation for continuous multiple ranking and selection. In *Proceedings of the 2017 Winter Simulation Conference*. IEEE, Piscataway, NJ, 2161–2172.

[25] Matthias Poloczek, Jialei Wang, and Peter I. Frazier. 2016. Warm Starting Bayesian Optimization. In *Proceedings of the 2016 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, 770–781.

[26] Guodong Shao, Sanjay Jain, Christoph Laroque, Loo Hay Lee, Peter Lendermann, and Oliver Rose. 2019. Digital Twin for Smart Manufacturing: The Simulation Aspect. In *Proceedings of the 2019 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, 2085–2098.

[27] Eunhye Song and Barry L. Nelson. 2019. Input–Output Uncertainty Comparisons for Discrete Optimization via Simulation. *Operations Research* 67, 2 (2019), 562–576.

[28] Eunhye Song, Barry L. Nelson, and L. Jeff Hong. 2015. Input Uncertainty and Indifference-zone Ranking & Selection. In *Proceedings of the 2015 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, 414–424.

[29] Eunhye Song and Uday V. Shanbhag. 2019. Stochastic Approximation for simulation Optimization under Input Uncertainty with Streaming Data. In *Proceedings of the 2019 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, 3597–3608.

[30] James C Spall. 1992. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control* 37, 3 (1992), 332–341.

[31] S. Surjanovic and D. Bingham. 2013. Virtual Library of Simulation Experiments: Test Functions and Datasets. Retrieved November 7, 2022, from http://www.sfu.ca/~ssurjano.

[32] Kevin Swersky, Jasper Snoek, and Ryan P Adams. 2013. Multi-Task Bayesian Optimization. In *Advances in Neural Information Processing Systems*, C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger (Eds.), Vol. 26. Curran Associates, Inc.

[33] Ferenc Szidarovszky. 2004. Global stability analysis of a special learning process in dynamic oligopolies. *Journal of Economic Research* 9, 2 (2004), 175–190.

[34] Andrei Nikolaevich Tikhonov, Vasilii Akovlevich Arsenin, and Vladimir Kotliar. 1976. *Méthodes de resolution de problèmes mal posés*. Éditions Mir, Moscow. 202 pages. Traduit du russe par Vladimir Kotliar.

[35] Juan Ungredda, Michael Pearce, and Juergen Branke. 2022. Bayesian Optimisation vs. Input Uncertainty Reduction. *ACM Trans. Model. Comput. Simul.* (jan 2022). Just Accepted.

[36] Haowei Wang, Jun Yuan, and Szu-Hui Ng. 2018. Informational Approach to Global Optimization with Input Uncertainty for Homoscedastic Stochastic Simulation. In *Proceedings of the 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. 1396–1400.

[37] Keqi Wang, Wei Xie, Wencen Wu, Bo Wang, Jinxiang Pei, Mike Baker, and Qi Zhou. 2020. Simulation-Based Digital Twin Development for Blockchain Enabled End-to-End Industrial Hemp Supply Chain Risk Management. In *Proceedings of the 2020 Winter Simulation Conference*. IEEE Press, Piscataway, NJ.

[38] Yidan Wang, Eunhye Song, and Huanan Zhang. 2019. *A Simulation Study on Emergency Medical Service Operations at the Centre LifeLink*. Technical Report. The Pennsylvania State University. https://bit.ly/2WVEbFO.

[39] Di Wu, Yuhao Wang, and Enlu Zhou. 2022. Data-Driven Ranking and Selection Under Input Uncertainty. *Operations Research* (2022).

[40] Di Wu and Enlu Zhou. 2017. Ranking and selection under input uncertainty: A budget allocation formulation. In *Proceedings of the 2017 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, 2245–2256.

[41] Jingxu Xu, Peter W. Glynn, and Zeyu Zheng. 2020. Joint Resource Allocation for Input Data Collection and Simulation. In *Proceedings of the 2020 Winter Simulation Conference*. IEEE Press, Piscataway, NJ.

[42] Jie Xu, Edward Huang, Liam Hsieh, Loo Hay Lee, Qing-Shan Jia, and Chun-Hung Chen. 2016. Simulation optimization in the era of Industrial 4.0 and the Industrial Internet. *Journal of Simulation* 10, 4 (2016), 310–320.