OPERATIONS RESEARCH
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0030-364X | EISSN 1526-5463 | 00 | 0000 | 0001

INFORMS
DOI 10.1287/xxxx.0000.0000
© 0000 INFORMS

General Bounds and Finite-Time Improvement for the Kiefer-Wolfowitz Stochastic Approximation Algorithm

Mark Broadie, Deniz Cicek, Assaf Zeevi
Graduate School of Business, Columbia University, New York, NY 10027,
mnb2@columbia.edu, dcicek05@gsb.columbia.edu, assaf@gsb.columbia.edu

We consider the Kiefer-Wolfowitz (KW) stochastic approximation algorithm and derive general upper bounds on its mean-squared error. The bounds are established using an elementary induction argument and phrased directly in the terms of tuning sequences of the algorithm. From this we deduce the non-necessity of one of the main assumptions imposed on the tuning sequences in the Kiefer-Wolfowitz paper and essentially all subsequent literature. The optimal choice of sequences is derived for various cases of interest, and an adaptive version of the KW algorithm, scaled-and-shifted KW (or SSKW), is proposed with the aim of improving its finite-time behavior. The key idea is to dynamically scale and shift the tuning sequences to better match them with characteristics of the unknown function and noise level, and thus improve algorithm performance. Numerical results are provided which illustrate that the proposed algorithm retains the convergence properties of the original KW algorithm while dramatically improving its performance in some cases.

Key words: stochastic optimization, stochastic approximation, the Kiefer-Wolfowitz algorithm, mean-squared-error convergence, finite-time improvement
History: This paper was first submitted on February 3, 2009. It is then revised on September 21, 2009 and March 16, 2010.

1. Introduction

Background and motivation. The term stochastic approximation refers to a broad class of optimization problems in which function values can only be computed in the presence of noise. Representative examples include stochastic estimation of a zero crossing, first introduced in the work of Robbins and Monro (1951), and stochastic estimation of the point of maximum, first studied by Kiefer and Wolfowitz (1952). Such problems arise in a variety of fields including engineering, statistics, operations research and economics, and the literature on the topic is voluminous; cf. the survey paper by Lai (2003) and the book by Kushner and Yin (2003).

A natural setting in which one encounters the need for stochastic approximation algorithms is simulation-based optimization. Here it is only possible to evaluate a function by means of simulation, and the observation noise is a direct consequence of the sample generating scheme; see, for example, Andradóttir (1995, 1996) for further discussion.

For concreteness we focus in this paper on the problem of sequential estimation of the point of maximum of an unknown function from noisy observations, noting that the main ideas developed in the paper extend in a straightforward manner to Robbins-Monro (RM) type algorithms; more specific commentary will be given in §2 and the electronic companion to this paper. In particular, we consider the following stochastic approximation scheme first studied by Kiefer and Wolfowitz (1952):

$$X_{n+1} = X_n + a_n \left(\frac{\widetilde{f}(X_n + c_n) - \widetilde{f}(X_n - c_n)}{c_n} \right), \quad n = 1, 2, \ldots \tag{1}$$

Here X_1 is the initial condition (either deterministic or random), $\{a_n\}$ and $\{c_n\}$ are two real-valued, deterministic *tuning sequences*, and $\widetilde{f}(X_n + c_n)$, $\widetilde{f}(X_n - c_n)$ are drawn according to conditional distribution functions $H(y|X_n + c_n)$ and $H(y|X_n - c_n)$ which have uniformly bounded second moments. Assuming the regression function $f(x) := \int y dH(y|x)$ admits a unique point of maximum and is strongly concave, Kiefer and Wolfowitz (1952) proved that the sequence $\{X_n\}$ generated by recursion (1) converges in probability to x^*, the unique maximizer of $f(\cdot)$, if $\{a_n\}$ and $\{c_n\}$ satisfy the following conditions:

(KW1) $c_n \to 0$ as $n \to \infty$;
(KW2) $\sum_{n=1}^{\infty} a_n = \infty$;
(KW3) $\sum_{n=1}^{\infty} \frac{a_n^2}{c_n^2} < \infty$;
(KW4) $\sum_{n=1}^{\infty} a_n c_n < \infty$.

Shortly after the publication of the KW algorithm, Blum (1954a) established that condition (KW4) is not necessary for convergence, leaving conditions (KW1)-(KW3) which have been imposed in almost all subsequent papers published on the subject (cf. Kushner and Yin (2003, §5.3.3) for a discussion of more general convergence conditions albeit in a more restricted setting). Roughly speaking, to have a convergent algorithm one requires that: (i) the gradient estimate localizes, hence c_n should shrink to zero; (ii) the step-size sequence a_n should shrink to zero, but in a manner that allows the algorithm to "cover" any distance from the initial point X_1 to the point of maximum, hence $\sum_n a_n$ diverges. If one adds the assumption that $a_n \to 0$ to (KW1) and (KW2), the role of (KW3) becomes questionable, and in fact, as this paper shows, superfluous.

A major focus in the literature has been establishing bounds on the mean-squared error (MSE) $\mathbb{E}|X_n - x^*|^2$, and deriving optimal rates at which the MSE converges to zero, under various assumptions on the unknown function and various modifications to the basic KW scheme; see, e.g., Derman (1956), Dupac (1957), Fabian (1967), Tsybakov and Polyak (1990). A common thread in these papers is that they all rely on a key lemma by Chung (1954) which restricts the tuning sequences $\{a_n\}$ and $\{c_n\}$ to be polynomial-like, specifically, of the form n^{-a} and n^{-c}, respectively, for some $a, c > 0$ such that conditions (KW1)-(KW3) hold. (Exceptions to this can be found in a stream of literature that develops weak convergence results; see, e.g., Burkholder (1956), Sacks (1958), and more recently Mokkadem and Pelletier (2007) as well as references therein.)

At a more practical level, the KW algorithm, theoretical convergence guarantees notwithstanding, has often been noted to exhibit poor behavior in implementations. The main culprit seems to be the tuning sequences which may not match up well with the characteristics of the underlying function. Hence there is a need to *adapt* the choice of these sequences to observed data points. Among the first to tackle this issue was Kesten (1958), who proposed a simple scheme to determine the step size at the nth iteration using the total number of sign changes of $\{X_m - X_{m-1} : m = 1, \ldots, n\}$. In a more recent paper, Andradóttir (1996) observed divergence of the KW algorithm when applied to functions which are "too steep," and proposed to adjust for this using two independent gradient estimates at each iteration.

A related issue arises when the magnitude of the step size is "too small" relative to the curvature of the function, which may lead to a degraded rate of convergence; see Nemirovski et al. (2009) for a simple example of this phenomenon. Ruppert (1988), Polyak (1990) and Polyak and Juditsky (1992) introduced the idea of *iterate averaging* to tackle this issue and proved that it guarantees asymptotically optimal convergence rates. Dippon and Renz (1997) use the same idea to propose a weighted averaging scheme specifically for the KW algorithm; see also further discussion in §3.

The convergence theory and specification of tuning sequences subject to (KW1)-(KW3) hinges on the *global* strong concavity/convexity of the underlying function $f(\cdot)$; see conditions (F1) and (F2) in §2. This assumption is unrealistic when it comes to most application settings. Kiefer and Wolfowitz (1952) identified this issue in their original paper, and proposed to "localize" the algorithm by restricting attention to a compact set (say, a closed bounded interval) which is known to contain the point of maximum. They argued that by projecting the iterates of the KW algorithm so that there will be no function evaluations outside of this set, one preserves the desired convergence properties without the need for the function to satisfy overly restrictive global regularity conditions. This *truncated* KW algorithm solves the divergence problem identified by Andradóttir (1996), however it introduces the problem of *oscillatory behavior* of the iterates: if the magnitude of the step-size sequence ($\{a_n\}$) is chosen too large relative to the magnitude of the gradient, the algorithm may end up oscillating back and forth between the boundaries of the truncation interval (see further discussion in §3). Andradóttir (1995) proposed an algorithm that adaptively determines the truncation interval, but still points to the oscillatory behavior as an open problem (see also Chen et al. (1999)). Finally, poor performance is also observed when function evaluations tend to be "too noisy," degrading the quality of the gradient estimate (see Vaidya and Bhatnagar (2006) who propose to replace the gradient estimate with its sign in order to mitigate this effect).

Main contributions. This paper makes contributions along the two dimensions discussed above. On the theoretical end, we present a new induction-based approach to bounding the MSE of the KW algorithm. The proof is simpler and more rudimentary than most extant methods which rely on martingale arguments or tools from weak convergence (cf. Kushner and Yin (2003)), and at the same time yields general bounds that hold under broad assumptions on the tuning sequences; see Theorem 1. Our assumptions allow for more general sequences than the ones typically found in the literature (see, for example, Dippon (2003) and Spall (1992)), and cover cases in which the MSE converges yet the sequences violate *necessary* conditions for almost sure convergence of the algorithm as laid out, for example, in Chen et al. (1999). The proof technique can be easily applied also to multidimensional settings (e.g., the one in Blum (1954b)), randomized modifications of KW (e.g., the Simultaneous Perturbation Stochastic Approximation (SPSA) procedure of Spall (1992)), and root finding variants of the Robbins-Monro type; see further commentary following Theorem 1 and §4. The bounds demonstrate that assumption (KW3) is in fact *not necessary* for the MSE to converge to zero (see §2.2.2), and at the same time allow us to deduce the *optimal* choice of tuning sequences $\{a_n\}$ and $\{c_n\}$ for a variety of cases of interest. Unlike previous literature, we do not impose polynomial decay a priori, but rather show how this property is *derived* from minimizing the order of our general MSE bounds (see Proposition 1 and Proposition 3). Other settings such as quadratic-like functions (see Proposition 2) and functions that satisfy further smoothness assumptions (see Theorem 2) are discussed as well.

Building on qualitative insights and intuition gleaned from our proofs, we present an adaptive version of the KW algorithm and illustrate via several examples its improved finite-time behavior. The algorithm is based on adaptively *scaling* the magnitude of the tuning sequences values, as well as *shifting* the index set. In particular, the rate degradation stemming from a step size that is "too small" is addressed by adaptively scaling up the $\{a_n\}$ sequence by a multiplicative constant. The oscillatory behavior that is due to a "too large" step size is solved by adaptively shifting the index of the $\{a_n\}$ sequence. Finally, the issue related to "large" simulation/estimation error in function evaluations is addressed by adaptively scaling up the $\{c_n\}$ sequence values. The MATLAB implementation of the algorithm can be downloaded from http://www.columbia.edu/˜mnb2/broadie/research.html/.

Remainder of the paper. Section 2 gives the main theoretical results, and discusses some implications, in particular, the non-necessity of assumption (KW3). Section 3 describes our proposed adaptive algorithm. Section 4 discusses extensions of the proof technique to more general settings such as multidimensional KW-type algorithms that allow for randomized directions, specifically the SPSA algorithm. All proofs are given in Section 5, while the Appendix contains a detailed description of the new adaptive KW algorithm (SSKW). Extensions of the proof technique to the multidimensional RM and KW algorithms are given in the electronic companion.

2. Performance Bounds and Their Implications

2.1. Bounds on the mean squared error

Consider the recursion (1) in the previous section. Throughout the paper, we assume that

$$\sigma^2 := \sup_{x\in\mathbb{R}} \mathrm{Var}[\widetilde{f}(X_n + c_n) - \widetilde{f}(X_n - c_n)|X_n = x] < \infty. \tag{2}$$

Remark 1. **(Observation noise)** The setting we treat in this paper, requiring the unknown function to be a conditional expectation, namely $f(x) := \int y dH(y|x)$, with bounded variance as in (2), allows for certain dependencies in the observations (e.g., common random numbers in gradient estimation), non-homogeneous noise and non-additive noise structure. A common setting for stochastic approximation is one where, conditioned on x, $\widetilde{f}(x_i) = f(x_i) + \varepsilon_i$, where $\{\varepsilon_i\}$ is a sequence of independent and identically distributed (i.i.d.) random variables with mean zero and finite variance bounded by σ^2. In the additive noise setting, requiring the unknown function to be a conditional expectation essentially restricts the noise to be a martingale difference sequence (which many books and papers on the topic of stochastic approximation take as a primitive assumption).

For the function f to be maximized, we assume that:
(F1) There exist finite positive constants K_0 and K_1 such that $K_0|x - x^*| \le |f'(x)| \le K_1|x - x^*|$ for all $x \in \mathbb{R}$, and
(F2) $f'(x)(x - x^*) < 0$ for all $x \in \mathbb{R} \setminus \{x^*\}$.

Remark 2. **(Objective function)** Assumptions (F1) and (F2) are identical to those found in most of the literature and will be used in Theorem 1; cf. Dupac (1957) and Wasan (1969). Assumption (F1) imposes a linearly growing envelope on the gradient. In essence, it guarantees that the function does not have flat regions away from the point of maximum. Assumption (F2) requires the function to be increasing for $x < x^*$ and decreasing for $x > x^*$, i.e., it has a "well-separated" point of maximum.

The tuning sequences to be used in the algorithm, $\{a_n\}$ and $\{c_n\}$, are assumed to be positive and bounded, and for some finite positive constants A, τ_1 and τ_2 satisfy:
(S1) $a_n/c_n^2 \le (a_{n+1}/c_{n+1}^2)(1 + Aa_{n+1})$ for all $n \ge 1$.
(S2) $c_n^2 \le c_{n+1}^2(1 + Aa_{n+1})$ for all $n \ge 1$.
(S3) $a_n \to 0$ as $n \to \infty$.
(S4) either (i) $c_n^4/a_n \le \tau_1$ or (ii) $c_n^4/a_n \ge \tau_2$, for all $n \ge 1$.

Remark 3. **(Tuning sequences)** The sequences $a_n = \theta_a/n^a$ and $c_n = \theta_c/n^c$ for $0 < a \le 1$ and $c \ge 0$ satisfy (S1)-(S4), but unlike most of the literature referenced in §1, these assumptions do not constrain $\{a_n\}$ and $\{c_n\}$ to be polynomial-like. In particular, they allow for a much broader class of sequences, some simple examples being $a_n = \theta_a/n, c_n = \theta_c/\log(n)$ and $a_n = \log(\log(n+2))/n, c_n = \theta_c$ with θ_a and θ_c being finite positive constants. We also note that the assumption "for all $n \ge 1$" in (S1)-(S4) is made mainly for simplicity; with obvious changes it can be replaced by "for all n sufficiently large."

The following is the main result of this section.

Theorem 1. *Let $\{X_n\}$ be generated by the Kiefer-Wolfowitz stochastic approximation recursion given in (1) using $\{a_n\}$ and $\{c_n\}$ satisfying (S1)-(S4) with $A < 4K_0$. Then under assumptions (F1) and (F2),*

$$\mathbb{E}(X_{n+1} - x^*)^2 \le \begin{cases} C_1 a_n/c_n^2 & \text{if } c_n^4 \le \tau_1 a_n \\ C_2 c_n^2 & \text{if } c_n^4 \ge \tau_2 a_n \end{cases} \tag{3}$$

for all $n \ge 1$, where C_1 and C_2 are finite positive constants identified explicitly in (35) and (38), respectively.

Proof outline. We only sketch the key ideas here. The full proof is given in Section 5. First, using assumptions (F1) and (F2) we derive bounds on the finite difference approximation of the gradient; see (19) and (21). Second, using the KW recursion (1) we express $(X_{n+1}-x^*)^2$ as a function of X_n. Then, after some algebra, taking expectations and using gradient bounds we get the real-number recursion:

$$b_{n+1} \leq (1-4a_nK_0+8K_1^2a_n^2)b_n + 2K_1a_nc_n\sqrt{b_n} + 2\frac{a_n^2}{c_n^2}\sigma^2 + 2K_1^2a_n^2c_n^2. \tag{4}$$

where $b_n := \mathbb{E}(X_n-x^*)^2$.

Now, since $a_n \to 0$ as $n\to\infty$, $(1-4a_nK_0+8K_1^2a_n^2) < 1$ holds for all n suitably large and we eventually have a contraction in recursion (4). This ensures convergence of the mean-squared error to zero as $n\to\infty$. To derive bounds on the MSE we use a straightforward induction argument where assumptions (S1)-(S4) are required for the induction step. We first use assumptions (S1) and (S2) along with the induction hypothesis to identify the higher order terms; these turn out to be either C_1a_n/c_n^2 or $C_2c_n^2$. Then, to finish the proof, we rely on (S3) to show that all remaining terms are of lower order. (This step involves the study of the behavior of a certain quadratic equation given in (33).) Expressions for the constants C_1 and C_2 are identified explicitly as part of this analysis. Q.E.D.

Remark 4. **(Truncated KW algorithm)** Thereom 1 requires the assumptions (F1) and (F2) to hold globally, which can be quite restrictive. This issue is also addressed in Kiefer and Wolfowitz (1952) where they argue that it suffices to have assumptions (F1) and (F2) hold only on a compact interval $I_0=[l,u]$, that is known to contain the point of maximum for the asymptotic theory to be valid. They propose projecting iterate $n+1$ onto a "truncation interval" $I_{n+1}=[l+c_{n+1},u-c_{n+1}]$ at step n so that there will be no function evaluations outside the interval I_0 (we assume $c_n<(u-l)/2$ for all $n\geq 1$). Such truncated algorithms are commonly used in the literature; see Andradóttir (1995) and Nemirovski et al. (2009) and references therein for some examples.

Using the same notation of the recursion given in (1), the *"truncated KW algorithm"* uses the recursion

$$X_{n+1} = \Pi_{I_{n+1}}\left(X_n + a_n\left(\frac{\widetilde{f}(X_n+c_n)-\widetilde{f}(X_n-c_n)}{c_n}\right)\right) \tag{5}$$

where $\Pi_{I_{n+1}}(\cdot)$ denotes the Euclidean projection operator onto the truncation interval $I_{n+1} = [l+c_{n+1},u-c_{n+1}]$. The results of Theorem 1 still hold for the truncated KW algorithm. The proof follows the same lines of the proof of Theorem 1 using the contraction property of the Euclidean projection operator.1.

Remark 5. **(Error bounds for the maximum)** Using a simple Taylor expansion and assumption (F1), we can derive from Theorem 1 upper bounds on $f(x^*)-\mathbb{E}f(X_n)$. Specifically, we have

$$\begin{aligned} f(x^*)-f(X_n) &= |x^*-X_n|\cdot|f'(\xi_n)| \quad \text{for some } \xi_n\in(x^*,X_n) \\ &\leq K_1|x^*-X_n|\cdot|\xi_n-x^*| \\ &\leq K_1(X_n-x^*)^2, \end{aligned}$$

where the first inequality follows from (F1) and the second since $\xi_n\in(x^*,X_n)$. Taking expectations and applying Theorem 1 we get

$$f(x^*)-\mathbb{E}(f(X_n)) \leq \begin{cases} K_1C_1a_n/c_n^2 & \text{if } \ c_n^4\leq\tau_1a_n \\ K_1C_2c_n^2 & \text{if } \ c_n^4\geq\tau_2a_n, \end{cases} \tag{6}$$

where C_1,C_2,τ_1,τ_2 are defined in Theorem 1.

Remark 6. **(Multidimensional extensions)** The result in Theorem 1, and the proof that supports it, can be easily extended to certain multidimensional versions of the KW algorithm, e.g., that of Blum (1954b), with some obvious modifications to assumptions (F1) and (F2); see Theorem EC.2 in the electronic companion to this paper. The proof technique can also be applied to "random direction"-type algorithms such as SPSA, introduced by Spall (1992), and related variants (cf. Chen et al. (1999)), by simply exploiting the tower property of conditional expectations; this is illustrated in Theorem 3 of Section 4.

Remark 7. **(Extensions to root-finding problems)** Consider the setting described by Robbins and Monro (1951). The problem is to sequentially find the unique root x^* of $g(x)=\xi$ using $\widetilde{g}(\cdot)$ which are noisy observations of $g(\cdot)$. Robbins and Monro (1951) consider the following stochastic approximation scheme:

$$X_{n+1}=X_n+a_n(\xi-\widetilde{g}(X_n)) \quad n=1,2,\dots \tag{7}$$

Here, $\widetilde{g}(X_n)$ is drawn according to the conditional distribution function $H(y|X_n)$ with $g(x):=\int y dH(y|x)$. The function $g(x)$ is assumed to satisfy $(x-x^*)g(x)\ge K_0(x-x^*)^2$ and $\mathbb{E}\widetilde{g}(x)^2\le K_1(1+(x-x^*)^2)$ for all $x\in\mathbb{R}$ and for some finite positive constants K_0, K_1. These are the standard assumptions in the Robbins-Monro (RM) context, cf. Benveniste et al. (1990). For any step-size sequence $\{a_n\}$ that satisfies $a_n\le a_{n+1}(1+Aa_{n+1})$ for some positive constant A such that $A<K_0$, one can easily show that

$$\mathbb{E}(X_{n+1}-x^*)^2\le Ca_n, \qquad \text{for all } n\ge 1, \tag{8}$$

for some finite positive constant C which can be explicitly identified. The proof follows almost verbatim the proof in Theorem 1. As a straightforward corollary of result (8), we conclude that the assumption $\sum_{n=1}^{\infty}a_n^2<\infty$, imposed in the majority of the stochastic approximation root-finding literature, is not required to prove convergence of the MSE to zero. Section EC.1 in the electronic companion extends this result to the multidimensional RM algorithm, and contains the full statement of the theorem along with its proof.

2.2. Implications

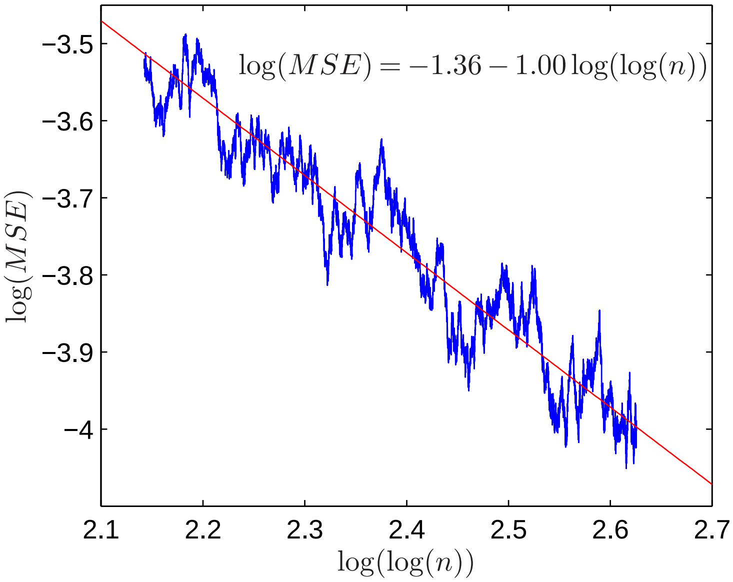
2.2.1. Optimizing the choice of tuning sequences From Theorem 1, it follows that $c_n\approx a_n^{1/4}$ minimizes the order of the upper bound on the MSE. With this choice Theorem 1 yields an MSE of order $\sqrt{a_n}$. This implies that one should choose $\{a_n\}$ to decrease as "fast" as possible while not violating (S1)-(S4). Proposition 1 shows that $a_n\approx 1/n$ is the optimal choice.

Proposition 1. *Let the assumptions of Theorem 1 hold and suppose $\{a_n\}$ is a non-increasing sequence. Then the minimal order of the upper bound in (3) is $O(1/\sqrt{n})$, which is achieved by setting $a_n=\theta_a/n$ and $c_n=\theta_c/n^{1/4}$ for any finite positive constants θ_a and θ_c with $\theta_a>(\sqrt{2}-1)/(2K_0)$.*

Remark 8. **(Optimality of polynomial-like sequences)** The result of Proposition 1 recovers the well known optimal rate of convergence of the KW algorithm under assumptions (F1) and (F2); see Dupac (1957) and Tsybakov and Polyak (1990). Unlike these papers, as well as essentially all antecedent literature, we *do not assume* the sequences to have the structure in the proposition, but rather *deduce* this structure from the more general bounds given in Theorem 1.

Remark 9. **(Specification and adjustment of the tuning sequences)** Once the optimal order of tuning sequences has been determined, it is then possible to optimize the constants θ_a and θ_c. In particular, if we possess a priori knowledge on the curvature of the function $f(\cdot)$ we can specify the sequence $\{a_n\}$ such that the condition $\theta_a>(\sqrt{2}-1)/(2K_0)$ holds, and hence ensure optimal convergence rates for the KW algorithm. Moreover, the explicit expressions for the constants in the upper bounds given in Theorem 1 can be used to further customize $\{a_n\}$ and $\{c_n\}$ so that these constants are optimized. In §3 we show how this idea leads to *adaptive* modifications of the KW algorithm that are applicable when one does not have good a priori knowledge of the function curvature, Lipschitz bounds, noise level, etc.

Figure 1 Illustration of the non-necessity of (KW3). The figure depicts the behavior of the MSE for a choice of sequences $\{a_n\}$ and $\{c_n\}$ that violates assumption (KW3); the MSE is seen to decay roughly like $(\log(n))^{-1}$, which follows from Theorem 1.



2.2.2. Non-necessity of (KW3) We exhibit sequences $\{a_n\}$ and $\{c_n\}$ which violate assumption (KW3), yet satisfy all assumptions of Theorem 1 and hence yield convergence of the mean-squared error to zero under the standard assumptions of (F1) and (F2). As mentioned in Remark 7, the non-necessity of (KW3) in the context of sequentially estimating the point of maximum translates into non-necessity of the assumption $\sum_{n=1}^{\infty} a_n^2 < \infty$ in the context of sequential root finding.

Put $a_n = 1/n$ and $c_n = \sqrt{\log(n+1)/n}$ for $n = 1, 2, \ldots$. It is easily verified that this choice satisfies (S1)-(S4). From Theorem 1 since $c_n^4 < \tau_1 a_n$ with $\tau_1 = 1$, we deduce that the MSE converges to zero at rate $O(a_n/c_n^2) = O(\log(n)^{-1})$ for any function satisfying assumptions (F1) and (F2) and such that $A < 4K_0$. At the same time, it is evident that $\sum_{n=1}^{\infty} a_n^2/c_n^2$ diverges, hence violating (KW3).

Figure 1 gives a plot of log(MSE) versus $\log\log(n)$ for this setting using the function $f(x) = x^2$. To find the MSE at each step, we run the algorithm 1000 times and average the results. The graph shows the results up to $n = 10^6$ steps of the algorithm. For numerical purposes, we assumed additive noise as described in Remark 1 using independent samples of a normal random variable with $\sigma = 1$ at each function evaluation. The regression coefficient in the log-log plot in Figure 1 is for iterations 5000 to 10^6 and is −1.0 (95% confidence interval $(-1.04, -0.97)$), consistent with Theorem 1 which for $a_n = 1/n$ and $c_n = \sqrt{\log(n)/n}$ predicts a convergence rate of $a_n/c_n^2 = 1/\log(n)$.

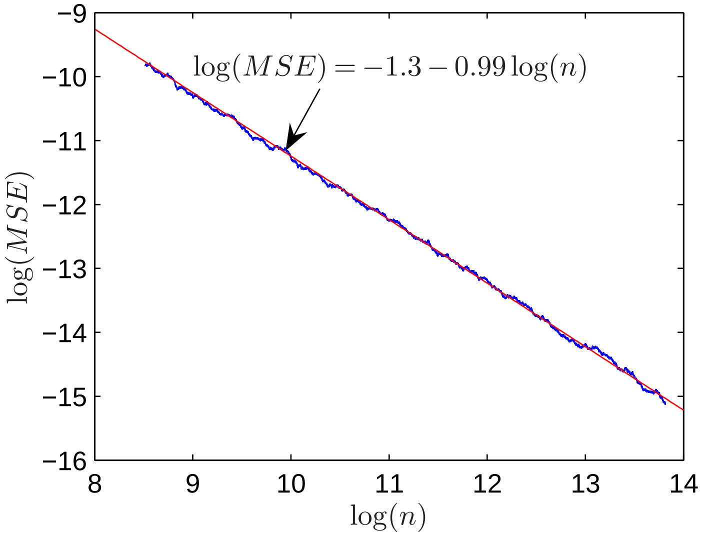
This choice of sequence only guarantees convergence of MSE and not almost sure convergence. Chen et al. (1999) develop necessary conditions for almost sure convergence of the iterates and this choice of sequence violates those conditions.

2.3. The special case of “quadratic-like” functions

The paper by Derman Derman (1956) analyzes the performance of the KW algorithm for the special case of quadratic-like functions, relying on Chung's lemma and hence restricting $\{a_n\}$ and $\{c_n\}$ to be polynomially decaying sequences. With this restriction the “best” rate of convergence for the MSE is shown to be $O(1/n^{1-\epsilon})$ for some $\epsilon > 0$. Next we revisit this analysis under the general framework developed in §2.

We first restate the assumption given in Derman (1956).

Figure 2 Illustration of non-necessity of (KW1) for "quadratic-like" functions. The figure plots $\log(MSE)$ versus $\log(n)$. The MSE behaves roughly like $O(n^{-1})$ which can be calculated using Proposition 2 with $a_n = 1/n$ and $c_n = 1$.



(F3) There exist positive constants K_0, K_1 and C_0 such that for every c, with $0 \le c \le C_0$,

$$-K_1(x-x^*)^2 \le \frac{f(x+c)-f(x-c)}{c}(x-x^*) \le -K_0(x-x^*)^2.$$

Proposition 2. *Let $\{X_n\}$ be generated by the KW recursion (1) using $\{a_n\}$ and $\{c_n\}$ that satisfy (S1) and (S3) with $A < 2K_0$. Then, under assumption (F3),*

$$\mathbb{E}(X_{n+1}-x^*)^2 \le Ca_n/c_n^2 \quad \text{for all } n \ge 1, \tag{9}$$

where C is identified explicitly in (48).

With this bound in place, we now exhibit the "best" choice of tuning sequences.

Proposition 3. *Let the assumptions of Proposition 2 hold and suppose $\{a_n/c_n^2\}$ is a non-increasing sequence. Then the minimal order of the upper bound in (9) is $O(1/n)$, which is achieved by setting $a_n = \theta_a/n$ and $c_n = \theta_c$ for any finite positive constants θ_a and θ_c satisfying $\theta_a > 1/K_0$.*

Unlike Proposition 1, here the finite-difference approximation of the gradient matches the true gradient for any value of c_n due to the "quadratic-like" function structure. As a result, the tradeoff between the two tuning sequences that determines the convergence rate in Theorem 1 does not exist in the setting of Proposition 3. Therefore, the optimal MSE convergence rate is achieved by setting the $\{c_n\}$ sequence to a constant value, in violation of condition (KW1). As a side note, by not relying on Chung's lemma we allow for more general sequences and use that to improve on the results of Derman (1956), eliminating the "for some $\varepsilon > 0$" in his convergence result.

To illustrate this numerically, let $a_n = 1/n$ and $c_n = 1$. This choice satisfies (S1) with $A = 2$ and (S3) with $\kappa = 1$. By Proposition 2, for any quadratic function, we should observe MSE convergence rate of order $1/n$. In particular for $f(x) = -x^2$, using additive independent standard normal noise at each function evaluation, Figure 2 plots $\log(MSE)$ versus $\log(n)$ up to $n = 10^6$ steps in the algorithm. The regression coefficient for this example is -0.99 (95% confidence interval $(-0.997, -1.005)$), which is close to the theoretical value of -1 predicted by Proposition 3.

2.4. Performance of the KW algorithm under further smoothness assumptions

Dupac (1957) derives the optimal rate of convergence for the basic KW algorithm (1) when the underlying function is thrice-differentiable. The result in Dupac (1957) is restricted to polynomial sequences as it again relies on the lemma by Chung (1954). We now revisit this problem and derive an analogue to Theorem 1.

We restrict our attention to functions that satisfy (F1), (F2) and:

(F4) $f'''(x)$ exists for all $x \in \mathbb{R}$ and $|f'''(x)| \leq T$ for some $T \in \mathbb{R}$.

For the sequences to be used in the algorithm we require (S1), (S3) and for some finite positive constants A, τ_1 and τ_2:

(S2') $c_n^4 \leq c_{n+1}^4(1 + Aa_{n+1})$ for all $n \geq 1$.

(S4') Either (i) $c_n^6/a_n \leq \tau_1$, or (ii) $c_n^6/a_n \geq \tau_2$, for all $n \geq 1$.

Remark 10. Since the functions are now assumed to be thrice-differentiable, we can expand to one further term in the Taylor expansion and derive a similar recursion to the one used to prove Theorem 1 (see proof sketch there). Hence we require assumptions (S2') and (S4') which replace (S2) and (S4) assumed in §2.

Theorem 2. *Let $\{X_n\}$ be generated by the Kiefer-Wolfowitz stochastic approximation recursion given in (1) with $\{a_n\}$ and $\{c_n\}$ satisfying (S1), (S2'), (S3) and (S4') with $A < 4K_0$. Then under assumptions (F1), (F2) and (F4)*

$$\mathbb{E}(X_{n+1} - x^*)^2 \leq \begin{cases} C_1' a_n/c_n^2 & \text{if } c_n^6/a_n \leq \tau_1 \\ C_2' c_n^4 & \text{if } c_n^6/a_n \geq \tau_2, \end{cases} \tag{10}$$

for all $n \geq 1$ and for some finite positive constants C_1' and C_2'.

The proof follows the same steps as in the proof of Theorem 1. The main difference is in the first step where we derive bounds on the gradient estimate using further smoothness assumed here. This adds one more term in the Taylor expansion of step 1 in the proof outline of Theorem 1, and in turn modifies the real number recursion for b_n outlined there.

Theorem 2 suggests that one should set $c_n \approx a_n^{1/6}$ to minimize the upper bound, whose order is then $O(a_n^{2/3})$. This implies that one should choose $\{a_n\}$ to decrease as "fast" as possible while not violating (S1), (S2'), (S3) and (S4') to get the optimal rate. The best choice of the tuning sequences is given as follows. The proof follows the same steps as the proof of Proposition 1, and hence we omit the details.

Proposition 4. *Let the assumption of Theorem 2 hold and suppose $\{a_n\}$ is a non-increasing sequence. Then the minimal order of the upper bound in (10) is $O(n^{-2/3})$, which is achieved by the setting $a_n = \theta_a/n$ and $c_n = \theta_c/n^{1/6}$ for any finite positive constants θ_a and θ_c that satisfy $\theta_a > (2^{2/3} - 1)/(2K_0)$.*

3. Finite-Time Behavior

3.1. Problems and remedies for finite-time behavior

Despite theoretical performance guarantees (e.g., those contained in Theorem 1), it is well known that stochastic approximation methods often perform quite poorly in practice. This emphasizes the importance of investigating the *finite-time behavior* of the algorithm, to complement the long run asymptotics and rates of convergence.

In this section we propose a modified version of the KW algorithm, which we call the *Scaled-and-Shifted KW algorithm.* This algorithm uses simple adaptive adjustments of the tuning sequences to address three main sources of poor performance:

1. a long oscillatory period due to a step-size sequence $\{a_n\}$ that is "too large;"

2. a degraded convergence rate due to a step-size sequence $\{a_n\}$ that is "too small;"

3. poor gradient estimates due to a gradient estimation step-size sequence $\{c_n\}$ that is "too small."

Next we explain in more detail each of these problems, illustrate them numerically and propose potential remedies that are combined in the final scaled-and-shifted KW algorithm.

3.1.1. The oscillation problem An issue that can arise in practical applications of the truncated KW algorithm (which is described in Remark 4) is a long period characterized by oscillations between boundaries of the truncation interval.

Definition 1. **(Oscillatory period)** Consider the truncated KW algorithm restricted to an interval $I_0 = [l, u]$. The oscillatory period T is defined as the number of iterations until the algorithm ceases consecutive visits to different boundary points, i.e.,

$$T = \sup\{n \geq 2 : (X_n = u - c_n \text{ and } X_{n-1} = l + c_{n-1}) \text{ or } (X_n = l + c_n \text{ and } X_{n-1} = u - c_{n-1})\}, \quad (11)$$

if the supremum on the right-hand-side above is finite, otherwise we set $T = 0$.

Roughly speaking, when the step-size sequence $\{a_n\}$ is too large relative to the gradient, the algorithm will exhibit a long transient period oscillating between boundary points until the step size becomes suitably small. This issue will not affect the algorithm's asymptotic performance, but the following example illustrates the severity of the problem. Figure 3(a) shows a single path of the truncated KW algorithm using $I_0 = [-50, 50]$ for the function $f(x) = -x^4$, and independent standard normal additive noise (i.e., $Y_i = f(x) + \varepsilon_i$, with $\varepsilon_i \sim N(0, \sigma^2)$ and $\sigma = 1$) and $X_1 = 30$. The tuning sequences are $a_n = 1/n$ and $c_n = 1/\sqrt[4]{n}$ as prescribed in §2. The oscillatory behavior can be observed for the first $T = 9960$ iterations and the algorithm only starts to converge after this period. The relative frequency of $X_{10,000}$ over many paths is illustrated in Figure 3(b). Even after 10,000 iterations, most of the paths are relatively far from $x^* = 0$.

The length $T = 9960$ of the oscillatory period depends on the length of the initial interval I_0. If one has more a priori information about the point of maxima and can specify a smaller initial interval, then the oscillatory period will be shorter. Similarly, less a priori information requires a larger initial interval, which leads to a longer oscillatory period. Figure 4 exhibits the relation between the average length of the oscillatory period estimated over 1000 sample paths and the length of the initial interval for the function $f(x) = -x^4$.

The long oscillatory period is caused by a step-size sequence $\{a_n\}$ that is too large in comparison to the magnitude of the gradient. To avoid this, we propose to decrease the step size when necessary by *shifting* the $\{a_n\}$ sequence; i.e., redefining the sequence $\{a'_n\} := \{a_{n+\beta}\}$ for some positive integer β. Specifically, whenever an iterate X_n falls outside the truncation interval known to contain x^*, we calculate the minimum positive integer β so that using $a_{n+\beta}$ ensures that the function evaluations are within the interval; i.e., both $X_n \pm c_n \in [l, u]$. The shifted sequence is used in the computation of all future iterates. Multiple shifts can occur, but the number of shifts is bounded in advance. Note that the shift(s) is adaptive, i.e., it is determined during the course of the algorithm and it does not require any additional information about the function. Figure 3(c) presents a typical sample path that results from applying the shift using the same parameters and random numbers as in Figure 3(b) and Figure 3(d) gives the relative frequency chart for $X_{10,000}$ using 1000 simulation replications.

Remark 11. **(Intuition for shifting)** The idea of shifting the $\{a_n\}$ sequence is inspired by close examination of the constants present in the upper bounds developed in §2. For instance, if we seek to minimize the constant C in the upper bound for "quadratic-like" functions, see (48) in Section 5, it is seen that this is achieved by balancing two terms. The first *decreases* with a decrease in the $\{a_n\}$ sequence for large values of K_1. The second term *increases* as $\{a_n\}$ decreases, so there is an

Figure 3 Oscillatory behavior of the truncated KW algorithm. Panel (a) shows a sample path of iterates in the truncated KW algorithm for the function $f(x) = -x^4$ with $a_n = 1/n$, $c_n = 1/\sqrt[4]{n}$ and $\sigma = 1$. The initial interval is assumed to be $I_0 = [-50, 50]$ and oscillatory behavior is observed for $T = 9960$ iterations. Panel (c) shows a sample path in the scaled-and-shifted KW algorithm in the same setting, using the same noise random sequence. The shift of $\beta = 9800$ corresponding to $a_n = 1/(n + 9800)$ is finalized after 28 iterations. The sequence $c_n = 1/\sqrt[4]{n}$ is not shifted. Panels (b) and (d) give the relative frequency of $X_{10,000}$ using 1000 simulation replications for the truncated KW and scaled-and-shifted KW algorithms, respectively.

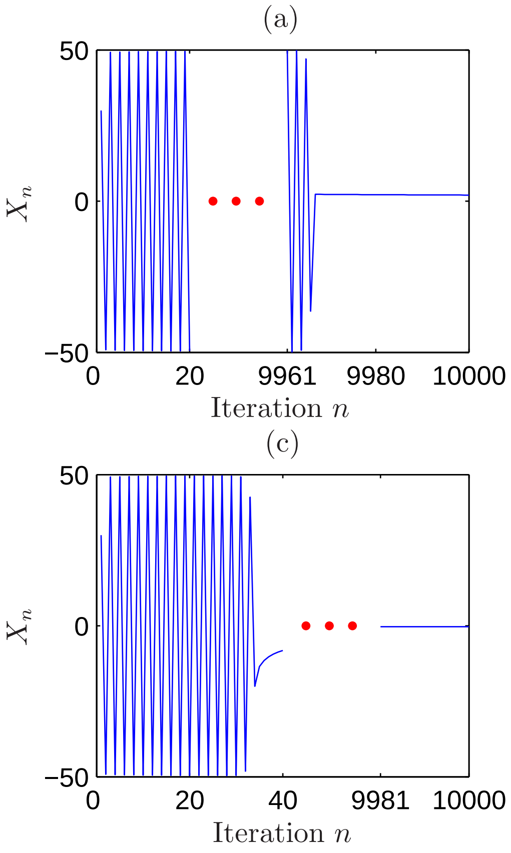
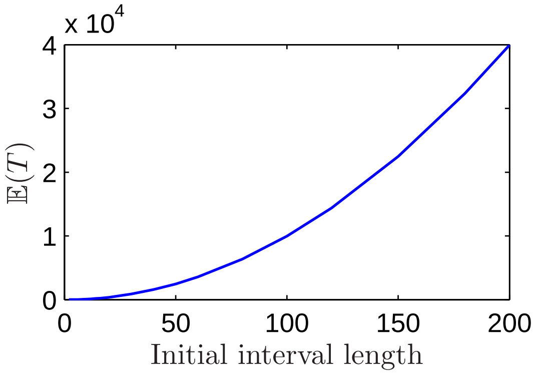


Figure 4 Oscillatory period vs. initial interval. This figure shows the estimated average length of the oscillatory period T as a function of the initial interval length $u - l$, for the function $f(x) = -x^4$ with $a_n = 1/n$, $c_n = 1/\sqrt[4]{n}$ and $\sigma = 1$.



evident tradeoff. The key observation is that when the gradient is steep, the first term dominates the second one and therefore a smaller $\{a_n\}$ sequence decreases the value of the constant C in our bound. Decreasing the step-size sequence $\{a_n\}$ by a shift preserves more "energy" for future iterations, since it does not dampen the entire subsequent entries in $\{a_n\}$ by the same multiplicative factor.

3.1.2. Degraded convergence rate due to a small step size The asymptotic results developed in the literature, as well as the bounds given in Theorem 1, require a careful choice of the $\{a_n\}$ sequence in relation to the curvature of the function that is being optimized. This is encoded in assumptions (S1) and (S2) with the requirement that $A < 4K_0$; see also Nemirovski et al. (2009) for further discussion. If the tuning sequences do not satisfy this assumption, for instance if the multiplicative constant θ_a in $a_n = \theta_a/n$ is not large enough, a degraded convergence rate may result. As a simple example, similar to the one worked out in Nemirovski et al. (2009), consider $f(x) = -0.001x^2$ with $a_n = 1/n$ and $c_n = 1/\sqrt[4]{n}$, and there is no observation error (i.e., $\sigma = 0$). Then the KW recursion becomes $X_{n+1} = X_n(1 - 1/(250n))$. Starting with $X_1 = 30$, we have

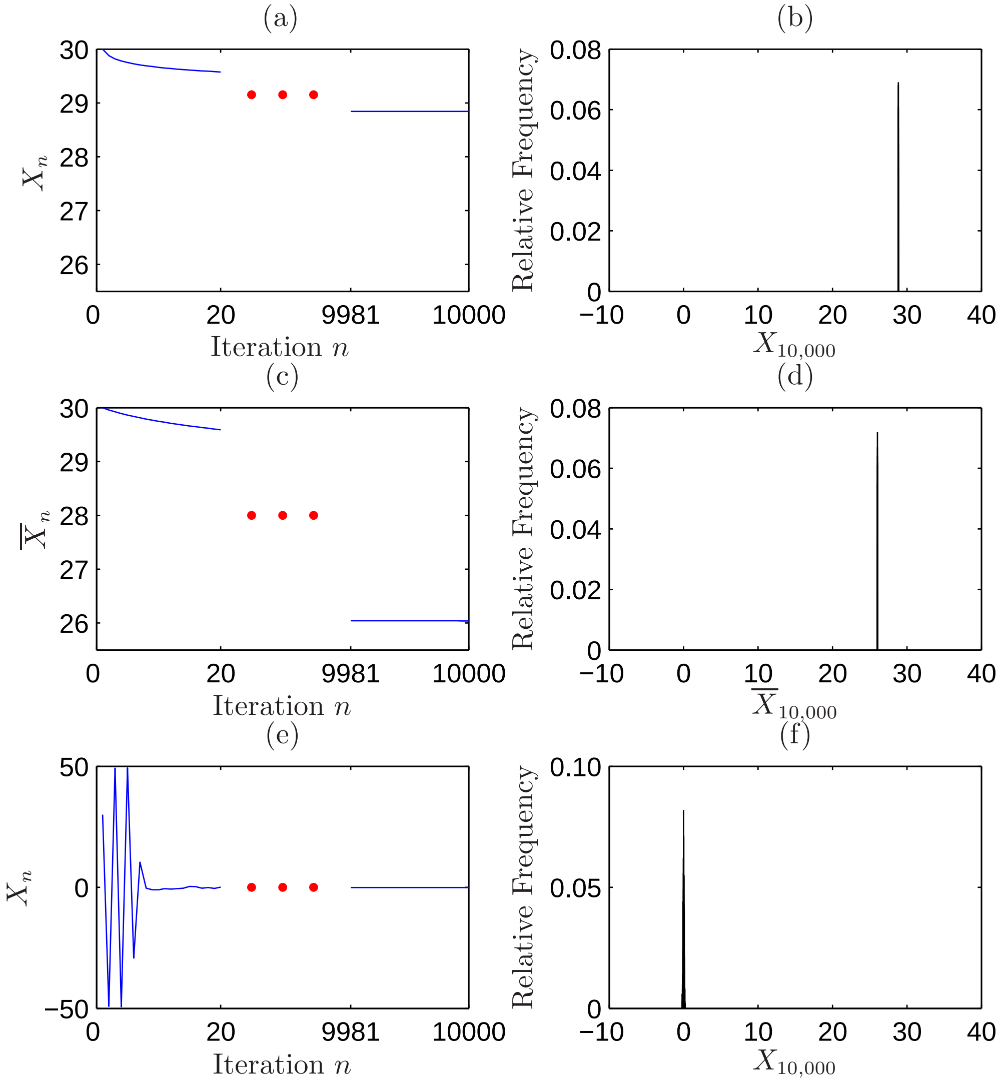
$$X_n = 30\prod_{m=1}^{n-1}\left(1 - \frac{1}{250m}\right) \geq \exp\left[-\sum_{m=1}^{n-1}\frac{1}{250m-1}\right] \geq \frac{27}{n^{0.004}}, \tag{12}$$

so the MSE cannot converge faster than $27^2/n^{0.008}$. In contrast, the upper bound in Theorem 1 guarantees a rate of $1/\sqrt{n}$, but this rate is not achieved because the $\{a_n\}$ sequence violates (S1) and (S2). Figure 5(a) illustrates a sample path of the iterates X_n in this setup with independent normal noise with zero mean and standard deviation $\sigma = 0.001$. The MSE convergence rate for this setting is -0.008 (see Table 3 for a corresponding confidence interval) which matches the theoretical rate given in (12). The relative frequency of $X_{10,000}$ given in Figure 5(b) shows all sample paths exhibit a similar lack of convergence.

The problem of degraded convergence rate due to the constant θ_a in $a_n = \theta_a/n$ being too small relative to the magnitude of the gradient is present both in the Robbins-Monro and the Kiefer-Wolfowitz algorithms. In order to tackle this problem in the RM framework, Ruppert (1988) and Polyak (1990) introduced the idea of averaging the iterates. They choose the $\{a_n\}$ sequence to converge to zero slower than $1/n$ and define $\overline{X}_n = \left(\sum_{i=1}^n X_i\right)/n$, where $\{X_n\}$ is the sequence generated by the RM algorithm with this choice of $\{a_n\}$ sequence. They prove that with these changes, $n^{1/2}(\overline{X}_n - x^*)$ converges in distribution to a normally distributed random variable with zero mean and variance that is independent of the constant in the $\{a_n\}$ sequence. Thus, the method achieves the optimal convergence rate for the RM framework independent of the choice of the constant in the tuning sequence. A corresponding result is developed by Dippon and Renz (1997) for the KW algorithm. In particular, for twice differentiable functions, the choice of $a_n = \theta_a\log(n)/n$ combined with iterate averaging guarantees the optimal convergence rate in the KW framework. This class of algorithms serve as a natural benchmark for our proposed algorithm.

Our remedy for this rate degradation problem is to scale up the $\{a_n\}$ sequence as follows. In the first several iterations of the algorithm, we multiply the $\{a_n\}$ sequence by a constant greater than or equal to one, so that iterate n is at the boundary of the current truncation interval, i.e., $X_n = l + c_n$ or $X_n = u - c_n$. This scaling up forces the algorithm to oscillate between the endpoints of the truncation interval $I_n = [l + c_n, u - c_n]$. This maps the problem of rate degradation into a problem of oscillatory behavior, which is then remedied by the shifted sequence approach of §3.1.1. The maximum number of forced boundary hits is a user-specified parameter set to four in all of our numerical experiments.2. Figure 5(e) shows a sample path of iterates generated by the scaled-and-shifted KW algorithm on $f(x) = -0.001x^2$ using the same parameters and same random numbers as in Figure 5(a). In this example, no shifting is needed after the $\{a_n\}$ sequence is scaled up and the optimal rate of convergence is recovered with this simple scaling (see Table 3 for

Figure 5 Degraded convergence rate due to a small step size. Panel (a) shows a sample path in the KW algorithm for $f(x) = -0.001x^2$ with $a_n = 1/n$, $c_n = 1/\sqrt[4]{n}$ and $\sigma = 0.001$. The $A < 4K_0$ assumption in (S1) and (S2) is violated. From Table 3, the convergence rate of the MSE is $-0.008 \pm 1.9 \times 10^{-6}$. Panel (b) is the relative frequency chart for $X_{10,000}$ exhibiting poor performance in all 1000 simulated sample paths. Panel (c) shows a sample path of Polyak-Ruppert averages of iterates generated in the exact same setting, but with using $a_n = \log(n)/n$. The MSE of the averages converges at a rate estimated to be $-0.05 \pm 1.6 \times 10^{-5}$. Relative frequency chart for $\overline{X}_{10,000}$ given in Panel (d) shows the poor convergence of the averages is present in all 1000 sample paths. Panel (e) shows a sample path in the scaled-and-shifted KW algorithm in the same setting using the same noise random sequence. After four scale-ups, the $\{a_n\}$ sequence becomes $a_n = 1987/n$ and shifting is not needed. From Table 3, the scaling results in an MSE convergence rate estimate of -0.53 ± 0.05 which recovers the optimal rate. As seen in Panel (f), this is observed in all of the 1000 simulated sample paths.



a confidence interval on the convergence rate). As seen in Figure 5(f), the scaled-and-shifted KW algorithm improves the convergence on all 1000 simulated samples although, unlike the Polyak-Ruppert scheme, there are no theoretical guarantees for the SSKW algorithm). The performance of Polyak-Ruppert averaging, with $a_n = \log(n+1)/n$, is displayed in Figure 5(c) under the same setting as in panel (a) and using identical random numbers. A slight improvement is noted relative to the TKW results given in panels (a) and (b) but the observed convergence behavior of Polyak-Ruppert averaging in this example is quite poor. In particular, the MSE exponent is calculated to be -0.05 (see Table 3 for a confidence interval) which is far from the guaranteed asymptotically optimal rate of -0.5. Figure 5(d) contains the relative frequency of final estimates and shows all 1000 sample paths exhibit similar poor performance.

3.1.3. The problem of noisy gradient estimates The finite-difference estimate of the gradient in (1) uses a tuning sequence $\{c_n\}$. Cases where the noise in the function observation is too large in magnitude relative to the $\{c_n\}$ sequence may give rise to excessive noise in the gradient estimates. As a consequence, even at the boundaries of the truncation interval, the algorithm may step away from the point of maximum of the function. Moreover, the iterates might move in random directions governed purely by the noise for a long period of iterations. This can lead to poor finite-time performance, even if the asymptotic convergence rate is eventually achieved. Figure 6(a) illustrates a sample path for the function $f(x)=1000\cos(\pi x/100)$ with $a_n=1/n$, $c_n=1/n^{1/4}$ and an initial interval $I_0=[-50,50]$. As before, we assume independent normal additive noise, i.e., $Y_i=f(x)+\varepsilon_i$, with $\varepsilon_i\sim N(0,\sigma^2)$ and $X_1=30$. The main difference is that we assume a large noise level given by $\sigma=1000$. The sample path in Figure 6(a) does not show convergent behavior for the first 10,000 iterations. (Similar behavior can be observed even up to 100,000 iterations.) The relative frequency of $X_{10,000}$ in Figure 6(b) shows a nearly uniform distribution between -50 and 50, i.e., the algorithm has not improved over X_1 in 10,000 iterations.

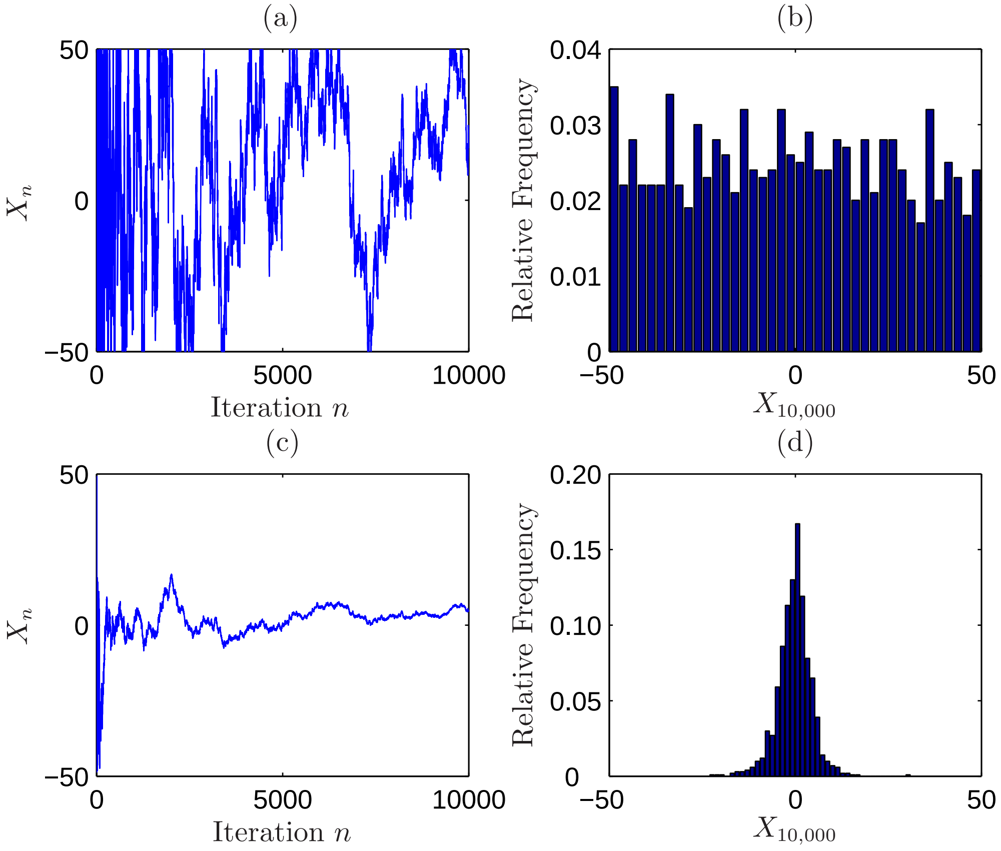
Our remedy for this problem is to scale up the $\{c_n\}$ sequence. Specifically, we multiply the $\{c_n\}$ sequence by a constant $\gamma_0>1$ when an iterate hits the boundary of the interval and the gradient estimate points in a direction away from the current truncation interval (i.e., away from x^*). This situation is one where the error in the gradient estimates is dominated by the noise term, since by assumption (F2) the true gradient at the boundary has to point towards x^*. We also make sure that the scaled-up $\{c_n\}$ sequence does not exceed an upper bound c_{max}, which is a parameter for our algorithm. In our numerical examples, we use $\gamma_0=2$. Multiple scale-ups can occur, but the number is bounded in advance. The scaled-up $\{c_n\}$ sequence is used for the remaining iterations of the algorithm. The $\{a_n\}$ sequence is also scaled and shifted as necessary as described before. Figure 6(c) shows the sample path of the scaled-and-shifted KW algorithm applied to the function $f(x)=1000\cos(\pi x/100)$ with the same parameters and random numbers. The $\{c_n\}$ sequence is scaled up four times at early stages of the algorithm, while the $\{a_n\}$ sequence is neither shifted nor scaled. With this adaptive tuning of the sequences, the iterates move toward the point of maximum much faster and this behavior is consistent throughout 1000 sample paths as shown in Figure 6(d). In this setting, the scaled-and-shifted KW algorithm achieves an MSE convergence rate of -0.53 ± 0.02 (see Table 5).

3.2. Numerical results

In this section we provide numerical results for the scaled-and-shifted KW algorithm, as described in the Appendix, which combines the remedies described previously. Results for the truncated KW algorithm are given for comparison. We also provide the results for Polyak-Ruppert averaging for the second example below that illustrate the rate degradation problem (since that scheme is only aimed at mitigating this particular issue). Algorithm sample paths are generated for 10,000 iterations. The standard errors are within 7% of the MSE values in all cases. Empirical convergence rates are calculated by computing a least squares fit of $\log(MSE)$ vs. $\log(n)$ using iterations $n=1000$ to $n=10{,}000$.

The initial tuning sequences are $a_n=1/n$ and $c_n=1/\sqrt[4]{n}$ in all cases but for the Polyak-Ruppert averaging case where $a_n=\log(n)/n$ is used (these are the settings proved to be optimal for the KW-algorithm in Proposition 1, and for Polyak-Ruppert averaging by Dippon and Renz (1997)).

Figure 6 Noisy gradient estimate problem. Panel (a) shows a sample path of the truncated KW algorithm for the function $f(x) = 1000\cos(\pi x/100)$ assuming an initial interval $I_0 = [-50, 50]$, using normally distributed noise with $\sigma = 1000$, $a_n = 1/n$ and $c_n = 1/n^{1/4}$. The MSE convergence rate estimate for this setting is -0.08 ± 0.008 (see Table 5). Panel (c) shows a sample path in the scaled-and-shifted KW algorithm for the same function, using the same random noise sequence. The algorithm adjusts the $\{c_n\}$ sequence to the noise level and shows much faster convergence. After scaling and shifting, the final sequences are $a_n = 1/n$ and $c_n = 16/n^{1/4}$. The last row in Table 5 corresponds to this setting and gives an estimate of the MSE rate of convergence of -0.53 ± 0.02. The relative frequency of $X_{10,000}$ in both algorithms are given in panels (b) and (d).



We report the statistics on the final adapted tuning sequences in separate tables. The initial interval used in all examples is $[-50, 50]$ and the initial starting point is always set to be $X_1 = 30$. The input parameters for the scaled-and-shifted KW algorithm are set so that the iterates hit the opposite boundaries of the truncation interval at the first four iterations, and the scale-up factor for the $\{c_n\}$ sequence is two. The upper bounds on the total number of shifts in the $\{a_n\}$ sequence and scale-ups in the $\{c_n\}$ sequence are both set to be 50. In order to prevent large shifts in a single iteration due to noise, the amount of shift in single iteration is upper bounded initially by 10 and every time a shift that equals to the upper bound is realized, we double this upper bound. Also to prevent a large scale-up in the $\{a_n\}$ sequence due to noise, the amount of scale-up per iteration is bounded by 10 in all runs. We also upper bound the $\{c_n\}$ sequence so that $c_n \leq 20$ for all $n \geq 1$. All functions are estimated using $\widetilde{f}(x_i) = f(x_i) + \varepsilon_i$ with independent noise $\varepsilon_i \sim N(0, \sigma^2)$.

Remark 12. **(Theoretical guarantees)** The result given in Theorem 1 also hold for the scaled-and-shifted KW algorithm. As mentioned in Remark 3, it is enough to have conditions (S1)-(S4) satisfied "for all sufficiently large n" for the bounds in Theorem 1 to hold. Since neither scaling nor shifting can occur more than finitely many times (the notation m_{max} introduced in the appendix), the conditions of Theorem 1 hold for all $n > m_{max}$. Together with the extension of Theorem 1 to the truncated KW algorithm (see Remark 4), this is sufficient to conclude that adaptation of the tuning sequences in the scaled-and-shifted KW algorithm preserve the theoretical performance guarantees, at least asymptotically.

Table 1 Comparison of the scaled-and-shifted KW algorithm and the truncated KW algorithm for $f(x) = -x^4$. This table shows the MSE calculated at iterations 100, 1000 and 10000, the convergence rate estimate for the MSE and the 5th and 95th percentile along with the median of the length of the oscillatory periods at different noise levels (σ). The numbers in square brackets [·] correspond to the truncated KW algorithm.

	MSE			Length of oscillatory period		
σ	100	1000	10000	5%	Median	95%
0.1	8.7	0.6	0.08	27	27	27
	[2469]	[2482]	[16]	[9960]	[9960]	[9960]
1	8.5	0.6	0.08	27	27	29
	[2469]	[2482]	[15]	[9959]	[9960]	[9960]
10	7.2	0.6	0.2	22	27	31
	[2469]	[2482]	[12]	[9957]	[9959]	[9961]

Table 2 Tuning sequence statistics for $f(x) = -x^4$. The 5th and 95th percentile along with the median values are given for the total scale-up factor α, and the total shift amount β, in the $\{a_n\}$ tuning sequence, as well as the total scale-up factor γ, for the $\{c_n\}$ sequence. For this test function, we observe only shifting of the $\{a_n\}$ sequence and no scaling up at all noise levels (σ).

	α			β			γ		
σ	5%	Median	95%	5%	Median	95%	5%	Median	95%
0.1	1	1	1	9799	9799	9799	1	1	1
1	1	1	1	9799	9799	9800	1	1	1
10	1	1	1	9796	9799	9801	1	1	1

Example 1. The first test function is $f(x) = -x^4$. This function does not satisfy assumption (F1) and hence we do not have a theoretical MSE convergence rate, but it serves to "stress test" the algorithm. When the truncated KW algorithm is applied to this function, slow convergence is often observed due to long oscillatory periods. Table 1 shows this effect and also shows that the scaled-and-shifted KW algorithm decreases the oscillatory period significantly for all noise levels and dramatically reduces the MSE which is calculated using 1000 independent replications. Statistics on the adaptations to the sequences are given in Table 2.

Example 2. The second test function is $f(x) = -0.001x^2$, which has a "flat" gradient away from the point of maximum. The $\{a_n = 1/n\}$ sequence then violates assumption $A < 4K_0$ of Theorem 1. This results in a degraded rate of convergence which also impacts the finite-time behavior of the algorithm. Table 3 shows that the estimated convergence rate of truncated KW algorithm is close to zero, i.e., it is not converging for all practical purposes. The Polyak-Ruppert averaging idea improves on this slightly, but the estimated convergence rate is still quite far from the optimal rate of -0.5. On the other hand, SSKW algorithm significantly improves the convergence behavior, recovering the optimal rate. Table 4 presents statistics on the adaptations to the sequences and shows that there is significant scaling up in the $\{a_n\}$ sequence. The scaling up in the $\{c_n\}$ sequence is more pronounced at high noise levels. Although there is no shifting in the $\{a_n\}$ sequence at small

Table 3 Comparison of the scaled-and-shifted KW algorithm, the truncated KW algorithm and the KW algorithm with Polyak-Ruppert (PR) averaging for $f(x) = -0.001x^2$. This table shows the MSE calculated at iterations 100, 1000 and 10000, the convergence rate for the MSE, and the 5th and 95th percentile and the median of the length of the oscillatory periods at different noise levels, σ.

		MSE			Convergence	Length of oscillatory period		
σ	Alg.	100	1000	10000	Rate	5%	Median	95%
0.001	SSKW	0.05	0.02	0.005	-0.51 ± 0.05	4	4	4
	PR	843	773	679	$-0.06 \pm 3.7 \times 10^{-6}$	0	0	0
	TKW	863	848	833	$-0.008 \pm 5.4 \times 10^{-7}$	0	0	0
0.01	SSKW	5.1	1.7	0.5	-0.51 ± 0.05	4	4	5
	PR	843	773	679	$-0.06 \pm 3.7 \times 10^{-5}$	0	0	0
	TKW	863	848	832	$-0.008 \pm 5.4 \times 10^{-6}$	0	0	0
0.1	SSKW	179	58	19	-0.50 ± 0.09	4	4	10
	PR	843	772	678	$-0.06 \pm 3.6 \times 10^{-4}$	0	0	0
	TW	863	847	832	$-0.008 \pm 5.3 \times 10^{-5}$	0	0	0
1	SSKW	243	73	24	-0.49 ± 0.1	4	4	11
	PR	844	784	696	-0.05 ± 0.003	0	0	0
	TKW	863	848	833	$-0.008 \pm 4.9 \times 10^{-4}$	0	0	0

σ values, when σ gets larger we observe occasional shifts as shown in the values for β. That is, the algorithm adjusts the magnitude of the $\{a_n\}$ sequence by shifting, if it was scaled up too much at the initial phase. All the numbers in Table 3 and Table 4 are calculated using 2000 independent replications.

Table 4 Tuning sequence statistics for $f(x) = -0.001x^2$. The 5th and 95th percentile along with the median values are given for the total scale-up factor α, and the total shift amount β, in the $\{a_n\}$ sequence, as well as the total scale-up factor γ, for the $\{c_n\}$ sequence, for various noise levels (σ).

	α			β			γ		
σ	5%	Median	95%	5%	Median	95%	5%	Median	95%
0.001	1968	2001	2035	0	0	0	1	1	1
0.01	1714	2007	2410	0	0	1	1	1	1
0.1	963	1794	7967	0	1	19	1	2	4
1	165	888	4187	0	2	15	4	16	32

Table 5 Comparison of the scaled-and-shifted KW algorithm and the truncated KW algorithm for $f(x) = 1000\cos(\pi x/100)$. This table shows the MSE calculated at iterations 100, 1000 and 10000, the convergence rate for the MSE, and the 5th and 95th percentile and the median of the length of the oscillatory periods at different noise levels (σ). The numbers in square brackets [·] correspond to the truncated KW algorithm.

	MSE			Convergence	Length of oscillatory period		
σ	100	1000	10000	Rate	5%	Median	95%
10	48	13	4	-0.50 ± 0.03	4	4	6
	[6]	[1.9]	[0.6]	$[-0.50 \pm 0.03]$	[0]	[0]	[0]
100	188	51	15	-0.52 ± 0.04	4	6	16
	[530]	[207]	[61]	$[-0.53 \pm 0.03]$	[0]	[0]	[3]
1000	252	79	24	-0.51 ± 0.06	4	6	16.5
	[1499]	[937]	[814]	$[-0.06 \pm 0.02]$	[29]	[61]	[114]

Example 3. The last test function is $f(x)=1000\cos(\pi x/100)$; this specification enables us to use the same truncation interval $[-50,50]$ used in the two other cases. Note that the function satisfies conditions (F1) and (F2) in the truncation interval. Table 5 shows that the scaled-and-shifted KW algorithm outperforms the truncated KW algorithm in both MSE and convergence rate measures for large noise levels. The only case where the truncated KW algorithm outperforms its adaptive counterpart in terms of MSE is at the lower noise level of $\sigma=10$. In this case, since the assumption $A<4K_0$ is satisfied for the initial choice of the $\{a_n\}$ sequence, the scaling up of the $\{a_n\}$ sequence decreases performance in terms of MSE. But since the algorithm does not "know" the assumption holds, it forces the iterates to hit the boundary at the first two iterations by increasing the step-size sequence $\{a_n\}$. Although the rate of convergence is still preserved, we observe slightly worse MSE results. Statistics about the adaptation of the sequences are given in Table 6. All the numbers in Table 5 and Table 6 are calculated using 3000 independent replications of the algorithm.

Table 6 Modifications in the tuning sequences for $f(x)=1000\cos(\pi x/100)$. The 5th and 95th percentile along with the median values are given for the total scale-up factor, α and the total shift amount, β in $\{a_n\}$ sequence as well as the total scale-up factor for the $\{c_n\}$ sequence, γ, for various noise levels (σ).

	α			β			γ		
σ	5%	Median	95%	5%	Median	95%	5%	Median	95%
10	4.5	6.9	15.1	0	0	7	1	1	1
100	1.8	7.3	33	0	4	37	2	8	16
1000	1.0	1.8	12	0	6	42	16	32	62

In all three examples, scaling and shifting the tuning sequences resulted in vastly improved finite-time performance, and essentially optimal estimates of the rate of convergence (for example,the improvement in the MSE can be as high as a factor of 150,000). In instances where the original choice of the sequences is a good fit to the characteristics of the underlying function, and where the TKW algorithm does seem to converge at the optimal rate, scaling and shifting does not degrade the convergence rate. In these instances (such as example 3 with $\sigma=10$ given in Table 5), the MSE values for the SSKW algorithm are slightly worse than those of the TKW algorithm mainly because of the scale-ups in the $\{a_n\}$ sequence which forces boundary hits.

4. Further Extensions

The induction-based proof technique that was introduced in the proofs of the main theoretical results can be used to establish upper bounds on the MSE of various other stochastic approximation algorithms; some of these extensions were indicated in Section 2. As further illustration, in this section, we use the same proof technique to establish upper bounds on the MSE of the SPSA algorithm introduced by Spall (1992); this is a representative example of a general class of multidimensional KW-type algorithms with randomized directions used in the finite-difference approximation of the gradient. These upper bounds are expressed directly in terms of the tuning sequences $\{a^{(n)}\}$ and $\{c^{(n)}\}$, as before. We establish convergence of the iterates to the true point of optimum x^* under more relaxed assumptions than the ones in Spall (1992) and to the best of our knowledge, under the most general assumptions found in related literature. As a reminder, for a d-dimensional stochastic approximation problem, the SPSA recursion is of the following form:

$$X_k^{(n+1)}=X_k^{(n)}+a^{(n)}\left(\frac{\widetilde{f}(X^{(n)}+\Delta^{(n)}c^{(n)})-\widetilde{f}(X^{(n)}-\Delta^{(n)}c^{(n)})}{\Delta_k^{(n)}c^{(n)}}\right), \tag{13}$$

for $k = 1, \dots, d$. Here $X_k^{(n)}$ denotes the k^{th} coordinate of the n^{th} iterate, $\widetilde{f}(x)$ denotes the noisy observation of the true function value, $f(x) = \mathbb{E}[\widetilde{f}(x)]$, and the sequences $\{a^{(n)}\}$ and $\{c^{(n)}\}$ are one-dimensional deterministic sequences as before. The sequences $\{\Delta_k^{(n)}\}$, $k = 1, \dots, d$ are i.i.d. for each n, bounded and symmetrically distributed around zero. As in Spall (1992), we assume that the observation noise is additive, i.e., $\widetilde{f}(x) = f(x) + \varepsilon$ for all $x \in \mathbb{R}^d$, where ε is independent of x and

$$\sigma^2 := \sup_{x \in \mathbb{R}^d} \operatorname{Var}\left(\widetilde{f}(x + c\Delta) - \widetilde{f}(x - c\Delta)|\Delta\right) \tag{14}$$

which allows for common random numbers in the two function evaluations required to estimate the gradient.

The assumptions on the perturbation sequence, $\Delta^{(n)} := (\Delta_1^{(n)}, \dots, \Delta_d^{(n)})^T$, are same as those in Spall (1992). Defining $\overline{\Delta^{(n)}} = (1/\Delta_1^{(n)}, \dots, 1/\Delta_d^{(n)})^T$, we assume

$$\mathbb{E}\left(\Delta_i^{(n)}\right) = 0, \quad \left|\Delta_i^{(n)}\right| \le \xi_1 \text{ for all } i = 1, \dots, d \quad \text{and} \quad \mathbb{E}\|\overline{\Delta^{(n)}}\|^2 \le \xi_2^2 \tag{15}$$

for all $n \ge 1$, where ξ_1 and ξ_2 are finite positive numbers.

For the underlying function, we relax the assumption $\mathbb{E}f(X^{(n)} \pm c^{(n)}\Delta^{(n)})^2 \le \alpha_1$ of Spall (1992). We also only assume that second derivatives exist and are bounded instead of the bounded third derivative assumption of Spall (1992). So the assumptions on the function are the following:
(F1') $(x - x^*)^T f'(x) \le -K_0\|x - x^*\|^2$
(F2') $f(\cdot)$ is twice differentiable with $|\partial^2 f(\cdot)/\partial x_i \partial x_j| \le K_1$ for all $i, j = 1, \dots, d$
for some finite positive constants $K_1 > K_0$.

As in Theorem 1, we prove the following theorem under assumptions (S1)-(S4) on the tuning sequences. So, we also relax the condition $\sum (a^{(n)}/c^{(n)})^2 < \infty$ assumed by Spall (1992).

Theorem 3. *Let $\{X^{(n)}\}$ be generated by the the recursion given in (13) using $\{a^{(n)}\}$ and $\{c^{(n)}\}$ satisfying (S1)-(S4) with $A < 2K_0$. Then under assumptions (F1'), (F2'), (14) and (15),*

$$\mathbb{E}(X^{(n+1)} - x^*)^2 \le \begin{cases} C_1'' a^{(n)}/(c^{(n)})^2 & \text{if } \ (c^{(n)})^4 \le \tau_1 a^{(n)} \\ C_2''(c^{(n)})^2 & \text{if } \ (c^{(n)})^4 \ge \tau_2 a^{(n)} \end{cases}$$

for all $n \ge 1$, where C_1'' and C_2'' are finite positive constants.

The proof is given in Section 5 and follows similar steps to the proof of Theorem 1. The main difference is in the step establishing upper bounds on the finite-difference estimate of the gradient, and the use of the tower property of conditional expectations handles the randomness due to the perturbation sequence $\{\Delta^{(n)}\}$.

Extension of the scaling and shifting ideas to the multidimensional setting is treated in a separate paper (Broadie et al. (2010)) which also contains numerical results on a set of realistic test problems spanning various fields.

5. Proofs

Proof of Theorem 1 **Step 1:** Fix $\{a_n\}$ and $\{c_n\}$ as in the statement of the theorem. For positive integer n and $x_n \in \mathbb{R}$, using Taylor expansion, there exist $0 \le T_1, T_2 \le 1$ such that

$$\begin{aligned} f(x_n + c_n) &= f(x_n) + f'(x_n + T_1 c_n)c_n \\ f(x_n - c_n) &= f(x_n) - f'(x_n - T_2 c_n)c_n. \end{aligned}$$

Using this, we have

$$\begin{aligned}\widehat{\nabla} f(x_n) &:= \frac{f(x_n+c_n)-f(x_n-c_n)}{c_n} \qquad (16)\\ &= f'(x_n+T_1c_n)+f'(x_n-T_2c_n)\\ &= \left[\frac{f'(x_n+T_1c_n)}{x_n-x^*+T_1c_n}+\frac{f'(x_n-T_2c_n)}{x_n-x^*-T_2c_n}\right](x_n-x^*)\\ &\quad +c_n\left[T_1\frac{f'(x_n+T_1c_n)}{x_n-x^*+T_1c_n}-T_2\frac{f'(x_n-T_2c_n)}{x_n-x^*-T_2c_n}\right]. \qquad (17)\end{aligned}$$

Now note that, using (F1) and (F2) we have

$$K_0 \le \left|\frac{f'(x)}{x-x^*}\right| = -\frac{f'(x)}{x-x^*} \le K_1. \qquad (18)$$

Using (17), we have

$$\begin{aligned}(x_n-x^*)\widehat{\nabla} f(x_n) &= \left[\frac{f'(x_n+T_1c_n)}{x_n-x^*+T_1c_n}+\frac{f'(x_n-T_2c_n)}{x_n-x^*-T_2c_n}\right](x_n-x^*)^2\\ &\quad +c_n(x_n-x^*)\left[T_1\frac{f'(x_n+T_1c_n)}{x_n-x^*+T_1c_n}-T_2\frac{f'(x_n-T_2c_n)}{x_n-x^*-T_2c_n}\right]\\ &\le -2K_0(x_n-x^*)^2+K_1c_n|x_n-x^*|, \qquad (19)\end{aligned}$$

where the inequality uses the fact that $f'(x)/(x-x^*) \le -K_0$ for any $x \in \mathbb{R}$, and the second term follows from

$$\begin{aligned}\left|T_1\frac{f'(x_n+T_1c_n)}{x_n-x^*+T_1c_n}-T_2\frac{f'(x_n-T_2c_n)}{x_n-x^*-T_2c_n}\right| &\le \max\left\{T_1\left|\frac{f'(x_n+T_1c_n)}{x_n-x^*+T_1c_n}\right|, T_2\left|\frac{f'(x_n-T_2c_n)}{x_n-x^*-T_2c_n}\right|\right\}\\ &\le \max\{T_1K_1, T_2K_1\}\\ &\le K_1. \qquad (20)\end{aligned}$$

Now, using the inequality $|a+b|^r \le 2^{r-1}(|a|^r+|b|^r)$ with (17), we obtain

$$\begin{aligned}[\widehat{\nabla} f(x_n)]^2 &\le 2\left[\frac{f'(x_n+T_1c_n)}{x_n-x^*+T_1c_n}+\frac{f'(x_n-T_2c_n)}{x_n-x^*-T_2c_n}\right]^2(x_n-x^*)^2\\ &\quad +2c_n^2\left[T_1\frac{f'(x_n+T_1c_n)}{x_n-x^*+T_1c_n}-T_2\frac{f'(x_n-T_2c_n)}{x_n-x^*-T_2c_n}\right]^2\\ &\le 8K_1^2(x_n-x^*)^2+2K_1^2c_n^2, \qquad (21)\end{aligned}$$

where the last inequality follows from bounding the first term using (18) and the second term using (20).

Step 2: Let X_n be the output of the n^{th} iterate of (1) and let $\widetilde{f}(X_n+c_n), \widetilde{f}(X_n-c_n)$ be the function observations at points X_n+c_n and X_n-c_n respectively. Note that

$$\mathbb{E}\left(\frac{\widetilde{f}(X_n+c_n)-\widetilde{f}(X_n-c_n)}{c_n}\;\Big|\; X_n\right) = \frac{f(X_n+c_n)-f(X_n-c_n)}{c_n} =: \widehat{\nabla} f(X_n), \qquad (22)$$

which together with the bounded variance assumption implies that

$$\begin{aligned}\mathbb{E}\left[\left(\frac{\widetilde{f}(X_n+c_n)-\widetilde{f}(X_n-c_n)}{c_n}\right)^2 \;\Big|\; X_n\right] &= \mathrm{Var}\left(\frac{\widetilde{f}(X_n+c_n)-\widetilde{f}(X_n-c_n)}{c_n}\;\Big|\; X_n\right)+[\widehat{\nabla} f(X_n)]^2\\ &\le \frac{\sigma^2}{c_n^2}+[\widehat{\nabla} f(X_n)]^2. \qquad (23)\end{aligned}$$

Now, using (1) we have

$$\begin{aligned} Z_{n+1} &:= (X_{n+1}-x^*)^2 = \left[X_n - x^* + a_n\left(\frac{\widetilde{f}(X_n+c_n)-\widetilde{f}(X_n-c_n)}{c_n}\right)\right]^2 \\ &= (X_n-x^*)^2 + 2a_n(X_n-x^*)\left(\frac{\widetilde{f}(X_n+c_n)-\widetilde{f}(X_n-c_n)}{c_n}\right) \\ &+ a_n^2\left(\frac{\widetilde{f}(X_n+c_n)-\widetilde{f}(X_n-c_n)}{c_n}\right)^2. \end{aligned} \tag{24}$$

Taking expectations of both sides conditioned on X_n and using (22) with (23) we get

$$\mathbb{E}(Z_{n+1}|X_n) \le Z_n + 2a_n(X_n-x^*)\widehat{\nabla}f(X_n) + a_n^2\left(\frac{\sigma^2}{c_n^2} + [\widehat{\nabla}f(X_n)]^2\right). \tag{25}$$

Using (19) and (21) we have

$$\mathbb{E}(Z_{n+1}|X_n) \le Z_n - 4a_nK_0Z_n + 2K_1a_nc_n\sqrt{Z_n} + \frac{a_n^2}{c_n^2}\sigma^2 + 8K_1^2a_n^2Z_n + 2K_1^2a_n^2c_n^2. \tag{26}$$

Finally, taking expectations, using the inequality $\mathbb{E}(\sqrt{Z_n}) \le \sqrt{\mathbb{E}(Z_n)}$, and setting $b_n := \mathbb{E}(Z_n)$ we get the following recursion:

$$b_{n+1} \le (1-4a_nK_0+8K_1^2a_n^2)b_n + 2K_1a_nc_n\sqrt{b_n} + \frac{a_n^2}{c_n^2}\sigma^2 + 2K_1^2a_n^2c_n^2. \tag{27}$$

Step 3: Before we start the induction proof, we will derive a crude upper bound on b_n that will be used later. Using $\sqrt{b_n} \le 1 + b_n$ in (27) we get

$$b_{n+1} \le b_n(1-4a_nK_0+8K_1^2a_n^2+2K_1a_nc_n) + 2K_1a_nc_n + \frac{a_n^2}{c_n^2}\sigma^2 + 2K_1^2a_n^2c_n^2,$$

which can be expressed more compactly as

$$b_{n+1} \le b_np_n + q_n, \tag{28}$$

with:

$$\begin{aligned} p_n &:= 1-4a_nK_0+8K_1^2a_n^2+2K_1a_nc_n > 0 \\ q_n &:= 2K_1a_nc_n + \frac{a_n^2}{c_n^2}\sigma^2 + 2K_1^2a_n^2c_n^2. \end{aligned}$$

Note that since $2K_1a_nc_n > 0$ by (S3), we have $p_n \ge 1-4a_nK_0+8K_1^2a_n^2$ which is a quadratic equation in a_n with positive leading coefficient, and $0 < K_0 < K_1$ ensures it has negative discriminant, hence $p_n > 0$.

Solving recursion (28), we get that for all n

$$b_n \le b_1\prod_{i=1}^{n}p_i + \sum_{i=2}^{n-1}q_i\prod_{j=i+1}^{n}p_j + q_n := B_n \tag{29}$$

which provides a crude upper bound on the MSE at the n^{th} step of the algorithm.

Put

$$n_0 := \sup\{n \geq 1 : (8K_1^2 - 4K_0 A)a_n + 8K_1^2 A a_n^2 \geq 4K_0 - A\} + 1, \tag{30}$$

and set $n_0 = 1$ if $(8K_1^2 - 4K_0A)a_n + 8K_1^2 A a_n^2 < 4K_0 - A$ for all n. Since we assume $A < 4K_0$, we have $n_0 < \infty$ because $a_n \to 0$ as $n \to \infty$ (assumption (S3)). Also, note that by (30)

$$\zeta := -\sup\{A - 4K_0 + (8K_1^2 - 4K_0A)a_n + 8K_1^2 A a_n^2 : n \geq n_0\} > 0. \tag{31}$$

Step 4: Now we will carry out the induction part of the proof.

Case (i): Suppose $c_n^4/a_n \leq \tau_1$, for all $n \geq 1$. We will first show that $b_{n+1} \leq C_1 a_n/c_n^2$ for all $n \geq n_0$ and some finite positive constant C_1. First, for $n = n_0$ suppose C_1 is chosen large enough to ensure $C_1 \geq B_{n_0+1}c_{n_0}^2/a_{n_0} \geq b_{n_0+1}c_{n_0}^2/a_{n_0}$. Now fix $n > n_0$ and suppose $b_{k+1} \leq C_1 a_k/c_k^2$ for all $n_0 \leq k \leq n-1$. We need to show that $b_{n+1} \leq C_1 a_n/c_n^2$. Using inequality (27) and the induction hypothesis we have

$$\begin{aligned} b_{n+1} &\leq (1 - 4a_nK_0 + 8K_1^2 a_n^2)C_1 \frac{a_{n-1}}{c_{n-1}^2} + 2K_1 a_n c_n \sqrt{C_1}\frac{\sqrt{a_{n-1}}}{c_{n-1}} + \frac{a_n^2}{c_n^2}\sigma^2 + 2K_1^2 a_n^2 c_n^2 \\ &\leq C_1 \frac{a_n}{c_n^2}(1 + Aa_n) - 4K_0 C_1 \frac{a_n^2}{c_n^2}(1 + Aa_n) + 8K_1^2 C_1 \frac{a_n^3}{c_n^2}(1 + Aa_n) \\ &+ 2K_1\sqrt{C_1}a_n^{3/2}(1 + \frac{A}{2}a_n) + \frac{a_n^2}{c_n^2}\sigma^2 + 2K_1^2 a_n^2 c_n^2, \end{aligned}$$

where for the second inequality we have used condition (S1) and the inequality $\sqrt{1 + Aa_n} \leq 1 + Aa_n/2$. Rearranging terms we get

$$\begin{aligned} b_{n+1} \leq C_1 \frac{a_n}{c_n^2} + \frac{a_n^2}{c_n^2} \Big\{ & C_1[A - 4K_0 + (8K_1^2 - 4K_0A)a_n + 8K_1^2 A a_n^2] \\ & + 2\sqrt{C_1}K_1(\frac{c_n^2}{\sqrt{a_n}} + \frac{A}{2}\sqrt{a_n}c_n^2) + \sigma^2 + 2K_1^2 c_n^4 \ \Big\}. \end{aligned}$$

Letting ν and κ denote the upper bounds on the $\{a_n\}$ and $\{c_n\}$ sequences, respectively, and using $c_n^2/\sqrt{a_n} \leq \sqrt{\tau_1}$, (31) gives:

$$b_{n+1} \leq C_1 \frac{a_n}{c_n^2} + \frac{a_n^2}{c_n^2}\left[-C_1\zeta + 2\sqrt{C_1}K_1(\sqrt{\tau_1} + \frac{A}{2}\sqrt{\nu}\kappa^2) + \sigma^2 + 2K_1^2\kappa^4\right]. \tag{32}$$

Now, if we can show that for some finite positive constant C_1,

$$-C_1\zeta + 2\sqrt{C_1}K_1(\sqrt{\tau_1} + \frac{A}{2}\sqrt{\nu}\kappa^2) + \sigma^2 + 2K_1^2\kappa^4 \leq 0, \tag{33}$$

then the induction proof would be complete. Viewing this as a quadratic in $\sqrt{C_1}$, we first observe that the leading coefficient is negative, by (31). It follows that this quadratic admits a solution, in particular, solving for the positive root and using $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$, we have $b_{n+1} \leq C_1 a_n/c_n^2$ for all $n \geq n_0$ with any choice of C_1 satisfying

$$C_1 \geq \max\left\{\left[\frac{2K_1(\sqrt{\tau_1} + \frac{A}{2}\sqrt{\nu}\kappa^2)}{\zeta} + \sqrt{\frac{\sigma^2 + 2K_1^2\kappa^4}{\zeta}}\right]^2, \frac{c_{n_0}^2}{a_{n_0}}B_{n_0+1}\right\}. \tag{34}$$

Finally let us modify the constant C_1 so that the result holds for all $n \geq 1$. This requires a simple modification in (34), and using $b_n \leq B_n$ can be done by setting

$$C_1 = \max\left\{\left[\frac{2K_1(\sqrt{\tau_1} + \frac{A}{2}\sqrt{\nu}\kappa^2)}{\zeta} + \sqrt{\frac{2\sigma^2 + 2K_1^2\kappa^4}{\zeta}}\right]^2, \max_{1\leq n\leq n_0}\left\{\frac{c_n^2}{a_n}B_{n+1}\right\}\right\}. \tag{35}$$

Case (ii): Suppose $c_n^4/a_n \geq \tau_2$, for all $n \geq 1$. Using similar steps to those in the proof of case (i), we will first show that $b_{n+1} \leq C_2 c_n^2$ for all $n \geq n_0$ for some finite positive constant C_2. First, for $n = n_0$ suppose C_2 is chosen large enough to assure $C_2 \geq B_{n_0+1}/c_{n_0}^2 \geq b_{n_0+1}/c_{n_0}^2$. Now suppose we have $b_{k+1} \leq C_2 c_k^2$ for all $n_0 \leq k \leq n-1$. We need to prove $b_{n+1} \leq C_2 c_n^2$. Using inequality (27) and the induction hypothesis we have

$$\begin{aligned} b_{n+1} &\leq (1 - 4a_n K_0 + 8K_1^2 a_n^2) C_2 c_{n-1}^2 + 2K_1 a_n c_n \sqrt{C_2} c_{n-1} + \frac{a_n^2}{c_n^2}\sigma^2 + 2K_1^2 a_n^2 c_n^2 \\ &\leq C_2 c_n^2 (1 + Aa_n) - 4K_0 C_2 a_n c_n^2 (1 + Aa_n) + 8K_1^2 C_2 a_n^2 c_n^2 (1 + Aa_n) \\ &+ 2K_1 \sqrt{C_2} a_n c_n^2 (1 + \frac{A}{2} a_n) + \frac{a_n^2}{c_n^2}\sigma^2 + 2K_1^2 a_n^2 c_n^2. \end{aligned}$$

where for the second inequality we have used (S2) with the inequality $\sqrt{1 + Aa_n} \leq 1 + Aa_n/2$. Rearranging terms we get

$$\begin{aligned} b_{n+1} &\leq C_2 c_n^2 + a_n c_n^2 \Big\{ C_2[A - 4K_0 + (8K_1^2 - 4K_0 A)a_n + 8K_1^2 A a_n^2] \\ &+ 2\sqrt{C_2} K_1 (1 + \frac{A}{2} a_n) + \sigma^2 \frac{a_n}{c_n^4} + 2K_1^2 a_n \Big\}. \end{aligned}$$

Using $a_n \leq \nu$, (31) and the assumption that $a_n/c_n^4 \leq 1/\tau_2$, we get

$$b_{n+1} \leq C_2 c_n^2 + a_n c_n^2 \left[-C_2 \zeta + 2\sqrt{C_2} K_1 (1 + \frac{A\nu}{2}) + \frac{\sigma^2}{\tau_2} + 2K_1^2 \nu \right]. \quad (36)$$

Similar to the first case, we need

$$-C_2 \zeta + 2\sqrt{C_2} K_1 (1 + \frac{A\nu}{2}) + \frac{\sigma^2}{\tau_2} + 2K_1^2 \nu \leq 0$$

for a suitable choice of C_2. Using the same argument as before, we have $b_{n+1} \leq C_2 c_n^2$ for all $n \geq n_0$ with any C_2 satisfying

$$C_2 \geq \max \left\{ \left[\frac{2K_1(1 + \frac{A\nu}{2})}{\zeta} + \sqrt{\frac{\sigma^2/\tau_2 + 2K_1^2 \nu}{\zeta}} \right]^2, \frac{1}{c_{n_0}^2} B_{n_0+1} \right\}. \quad (37)$$

Setting

$$C_2 = \max \left\{ \left[\frac{2K_1(1 + \frac{A\nu}{2})}{\zeta} + \sqrt{\frac{\sigma^2/\tau_2 + 2K_1^2 \nu}{\zeta}} \right]^2, \max_{1 \leq n \leq n_0} \left\{ \frac{1}{c_n^2} B_{n+1} \right\} \right\}. \quad (38)$$

we get the result for all $n \geq 1$ and this completes the proof.

Proof of Proposition 1 First, we claim that the optimal rate of convergence is achieved with sequences $\{a_n\}$ and $\{c_n\}$ that satisfy $c_n^4/a_n = \tau$. To see this note that if $c_n^4/a_n < \tau$, then we are in case (i) of Theorem 1 and the rate of convergence is a_n/c_n^2. But if we increase c_n or decrease a_n until we get $c_n^4/a_n = \tau$, we achieve a tighter bound. The same line of argument applies when $c_n^4/a_n > \tau$. Hence the best possible bound in (3) is of order $\sqrt{a_n}$. Thus, once we specify the $\{a_n\}$ sequence, we set $\{c_n\}$ such that $c_n^4/a_n = \tau$.

Next we show that $a_n = O(1/n)$ is the optimal order of magnitude for the $\{a_n\}$ sequence, in the sense that this choice yields the fastest convergence to zero of the MSE among those sequences satisfying assumptions (S1)-(S4). Now, clearly $a_n = \theta_a/n$ and $c_n = \theta_c/n^{1/4}$ satisfy assumptions (S1)-(S4). Suppose, towards a contradiction, that $\{a_n\}$ is of lower order than $1/n$, i.e., suppose

$a_n = \theta_a s_n/n$ for some finite positive sequence $\{s_n\}$ such that $s_n \to 0$ as $n \to \infty$ and $\{a_n\}$ is non-increasing.

We first observe that since $\{a_n\}$ is non-increasing, i.e., $a_{n+1} \leq a_n$, we have $s_{n+1} \leq s_n(n+1)/n$. Using this, we have

$$\frac{s_{n+1}^2}{n+1} \leq \frac{s_n^2}{n}\left(1+\frac{1}{n}\right) \leq 2\frac{s_n^2}{n}. \tag{39}$$

Now, note that (S1) and (S2) imply $a_n \leq a_{n+1}(1+\overline{A}a_{n+1})$ for some constant $\overline{A}$. Using this, we have $s_n/n \leq s_{n+1}/(n+1) + \overline{A}\theta_a s_{n+1}^2/(n+1)^2$, which implies

$$\begin{aligned} s_{n+1} &\geq s_n\left(1+\frac{1}{n}\right) - \overline{A}\theta_a \frac{s_{n+1}^2}{n+1} \\ &\geq s_n\left(1+\frac{1}{n} - 2\overline{A}\theta_a \frac{s_n}{n}\right), \end{aligned} \tag{40}$$

where we used (39) for the second inequality. Since $s_n \to 0$ as $n \to \infty$, we have that $2\overline{A}\theta_a s_n \leq 1/2$ for n sufficiently large. Using this in (40), we get $s_{n+1} \geq s_n(1+1/(2n))$ for all n sufficiently large. But this implies $s_n \to \infty$, which contradicts the assumption $s_n \to 0$ as $n \to \infty$. Therefore the optimal choice is $a_n = \theta_a/n$, and hence $c_n = \theta_c/n^{1/4}$ for positive constants θ_a and θ_c.

The last step is to find the restrictions on θ_a and θ_c to ensure that the condition $A < 4K_0$ holds. If we substitute these sequences in (S1) or (S2), after some algebra, we get

$$A \geq \frac{n+1}{\theta_a}\left(\sqrt{\frac{n+1}{n}} - 1\right).$$

Combining this with $A < 4K_0$, we get

$$\theta_a > \frac{1}{4K_0}\left[(n+1)\left(\sqrt{\frac{n+1}{n}} - 1\right)\right] \geq \frac{\sqrt{2}-1}{2K_0},$$

since the term in square brackets is maximized when $n = 1$.

Proof of Proposition 2 **Step 1:** Without loss of generality, we can assume $c_n \leq C_0$ for all $n \geq 1$ since we can scale down the sequence otherwise. Using (F3) and $\widehat{\nabla} f(x) = (f(x_n + c_n) - f(x_n - c_n))/c_n$ we have

$$(x - x^*)\widehat{\nabla} f(x) \leq -K_0(x-x^*)^2. \tag{41}$$

Squaring the terms in (F3), we obtain

$$[\widehat{\nabla} f(x)]^2 \leq K_1^2 (x-x^*)^2. \tag{42}$$

Step 2: We now derive a real number recursion for $b_n = \mathbb{E}(Z_n)$ using the same ideas as in Step 2 of the proof of Theorem 1. Using (25) and taking expectations of both sides and applying the bounds (41) and (42), denoting $b_n = \mathbb{E}(Z_n)$, we get

$$\begin{aligned} b_{n+1} &\leq b_n - 2a_n K_0 b_n + \frac{a_n^2}{c_n^2}\sigma^2 + K_1^2 a_n^2 b_n \\ &= b_n(1 - 2a_n K_0 + K_1^2 a_n^2) + \frac{a_n^2}{c_n^2}\sigma^2. \end{aligned} \tag{43}$$

Step 3: In the third step of the proof of Theorem 1, we established a bound for sequences satisfying $b_{n+1} \leq b_n p_n + q_n$. Using this, we have $b_n \leq B_n$, where B_n is defined in (29)and $p_n := 1 - 2a_n K_0 + K_1^2 a_n^2 > 0$, $q_n := \sigma^2 a_n^2/c_n^2$.

Now define

$$n_0' := \sup\{n \geq 1 : (K_1^2 - 2AK_0)a_n + K_1^2 A a_n^2 \geq 2K_0 - A\} + 1 \tag{44}$$

and set $n_0' = 1$ if $(K_1^2 - 2AK_0)a_n + K_1^2 A a_n^2 < 2K_0 - A$ for all $n \geq 1$. Using $A < 2K_0$, and since it is assumed in (S3) that $a_n \to 0$ as $n \to \infty$ we have $n_0' < \infty$. We note that by (44)

$$\zeta := -\sup\{A - 2K_0 + (K_1^2 - 2AK_0)a_n + K_1^2 A a_n^2 : n \geq n_0'\} > 0. \tag{45}$$

Step 4: We will again use induction to complete the proof. For $n = n_0'$ suppose C is chosen large enough to ensure $C \geq B_{n_0'+1} c_{n_0'}^2 / a_{n_0'} \geq b_{n_0'+1} c_{n_0'}^2 / a_{n_0'}$. Now suppose $b_{k+1} \leq C a_k / c_k^2$ for all $n_0' \leq k \leq n-1$. We need to show that $b_{n+1} \leq C a_n / c_n^2$. Using (43) and the induction hypothesis we have

$$\begin{aligned} b_{n+1} &\leq (1 - 2a_n K_0 + K_1^2 a_n^2) C \frac{a_{n-1}}{c_{n-1}^2} + \frac{a_n^2}{c_n^2}\sigma^2 \\ &\leq C\frac{a_n}{c_n^2}(1 + Aa_n) - 2K_0 C \frac{a_n^2}{c_n^2}(1 + Aa_n) + K_1^2 C \frac{a_n^3}{c_n^2}(1 + Aa_n) + \frac{a_n^2}{c_n^2}\sigma^2, \end{aligned}$$

where for the second inequality we have used condition (S1). Rearranging terms we get

$$b_{n+1} \leq C\frac{a_n}{c_n^2} + \frac{a_n^2}{c_n^2}\left\{ C(A - 2K_0 + (K_1^2 - 2K_0 A)a_n + K_1^2 A a_n^2) + \sigma^2 \right\}. \tag{46}$$

Using (45) gives

$$b_{n+1} \leq C\frac{a_n}{c_n^2} + \frac{a_n^2}{c_n^2}(-C\zeta + \sigma^2). \tag{47}$$

Using a similar argument to the one in proof of Theorem 1, with

$$C = \max\left\{\frac{\sigma^2}{\zeta}, \max_{1 \leq n \leq n_0'}\left\{\frac{c_n^2}{a_n}B_n + 1\right\}\right\}. \tag{48}$$

we have $b_{n+1} \leq C a_n / c_n^2$ for all $n \geq 1$ and this completes the proof.

Proof of Proposition 3 First note that with the choice of sequences and with $\theta_a > (\sqrt{2} - 1)/K_0$, we satisfy all the conditions of Proposition 2 (the inequality is verified at the end of the proof). Using this, the rate of convergence is a_n / c_n^2. To obtain the optimal rate of convergence, we should choose a_n / c_n^2 as small as possible such that it satisfies the assumptions in the proposition. Using assumption (S1), and $c_n \leq \kappa$ for all $n \geq 1$, we get

$$\frac{a_n}{c_n^2} \leq \frac{a_{n+1}}{c_{n+1}^2}(1 + Aa_{n+1}) \leq \frac{a_{n+1}}{c_{n+1}^2}(1 + A\kappa^2 \frac{a_{n+1}}{c_{n+1}^2}). \tag{49}$$

Substituting $d_n = a_n / c_n^2$ and $D = A\kappa^2$ in (49), to complete the proof, we need to find the non-increasing sequence $\{d_n\}$ satisfying $d_n \leq d_{n+1}(1 + Dd_{n+1})$ which converges to zero as fast as possible. But in the proof of Proposition 1, we showed that under these condition, the best choice of $\{d_n\}$, in the sense of minimizing the order of the MSE, is $d_n = \theta_d / n$ for some finite positive constant θ_d. This can be achieved by choosing $a_n = \theta_a / n$ and $c_n = \theta_c$ for some finite positive constants θ_a and θ_c. As we did in the proof of Proposition 1, the last step is to translate the requirement $A < 2K_0$ into a condition on θ_a. Substituting the sequences in assumption (S1) gives the condition $A \geq [(n+1)/n]/\theta_a$. With $A < 2K_0$, this requires $\theta_a > [(n+1)/n]/(2K_0)$ which completes the proof since the term in the square brackets is maximized when $n = 1$.

Proof of Theorem 3 Fix n and consider iterate $X^{(n)}$. First we define

$$\widehat{\nabla} f(X^{(n)}) := \left(\frac{f(X^{(n)} + \Delta^{(n)} c^{(n)}) - f(X^{(n)} - \Delta^{(n)} c^{(n)})}{c^{(n)}} \right) \overline{\Delta^{(n)}}$$

Using Taylor expansion, for some diagonal matrices $T_{\pm} \in \mathbb{R}^{d \times d}$ with all diagonal elements in $(0,1)$ and points $\overline{X^{(n)}_{\pm}}$ on the line segment between $X^{(n)}$ and $X^{(n)} \pm c^{(n)} \Delta^{(n)}$ we have,

$$\begin{aligned} \widehat{\nabla} f(X^{(n)}) &= (\Delta^{(n)})^T f'(X^{(n)}) \overline{\Delta^{(n)}} + \frac{c^{(n)}}{2} \left\{ (\Delta^{(n)})^T \left[f''\left(\overline{X^{(n)}_{+}}\right) - f''\left(\overline{X^{(n)}_{-}}\right) \right] \Delta^{(n)} \right\} \overline{\Delta^{(n)}} \\ &\leq (\Delta^{(n)})^T f'(X^{(n)}) \overline{\Delta^{(n)}} + K_1 d^2 \xi_1^2 c^{(n)} |\overline{\Delta^{(n)}}| \end{aligned} \tag{50}$$

where $f'(x)$ is the gradient vector and $f''(x)$ is the Hessian matrix at point x. Here the second inequality follows since, by assumption (F2") and (15),

$$(\Delta^{(n)})^T \left[f''\left(\overline{X^{(n)}_{+}}\right) - f''\left(\overline{X^{(n)}_{-}}\right) \right] \Delta^{(n)} \leq 2 K_1 d^2 \xi_1^2.$$

Using (50), we have

$$\begin{aligned} (X^{(n)} - x^*)^T \widehat{\nabla} f(X^{(n)}) \leq & \sum_{i=1}^{d} (X_i^{(n)} - x_i^*) \frac{\sum_{j=1}^{d} \Delta_j^{(n)} f_j'(X^{(n)})}{\Delta_i^{(n)}} \\ & + K_1 d^2 \xi_1^2 c^{(n)} \|X^{(n)} - x^*\| \|\overline{\Delta^{(n)}}\|. \end{aligned} \tag{51}$$

Taking expectations conditioned on $X^{(n)}$ in (51), we get

$$\begin{aligned} \mathbb{E}\left((X^{(n)} - x^*)^T \widehat{\nabla} f(X^{(n)}) | X^{(n)} \right) &\leq (X^{(n)} - x^*)^T f'(X^{(n)}) + K_1 d^2 \xi_1^2 c^{(n)} \|X^{(n)} - x^*\| \mathbb{E} \|\overline{\Delta^{(n)}}\| \\ &\leq -K_0 \|X^{(n)} - x^*\|^2 + K_1 d^2 \xi_1^2 \xi_2 c^{(n)} \|X^{(n)} - x^*\| \end{aligned} \tag{52}$$

where the first inequality holds since $\mathbb{E}\Delta_i^{(n)} = 0$ and the second follows using (F1') and the fact that (15) implies $\mathbb{E}\|\Delta^{(n)}\| \leq \xi_2$.

Using $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$ with (50) we get,

$$\begin{aligned} \|\widehat{\nabla} f(X^{(n)})\|^2 &\leq 2 \|(\Delta^{(n)})^T f'(X^{(n)}) \overline{\Delta^{(n)}}\|^2 + 2 K_1^2 d^4 \xi_1^4 (c^{(n)})^2 \|\overline{\Delta^{(n)}}\|^2 \\ &\leq 2 d \xi_1^2 \|f'(X^{(n)})\|^2 \|\overline{\Delta^{(n)}}\|^2 + 2 K_1^2 d^4 \xi_1^4 (c^{(n)})^2 \|\overline{\Delta^{(n)}}\|^2 \\ &\leq 2 K_1^2 d \xi_1^2 \|X^{(n)} - x^*\|^2 \|\overline{\Delta^{(n)}}\|^2 + 2 K_1^2 d^4 \xi_1^4 (c^{(n)})^2 \|\overline{\Delta^{(n)}}\|^2 \end{aligned}$$

where we use (15) for the second inequality and (F1') for the last one. Now, taking expectations conditioned on $X^{(n)}$ and using $\mathbb{E}\|\overline{\Delta^{(n)}}\|^2 \leq \xi_2^2$ we get

$$\mathbb{E}\left(\|\widehat{\nabla} f(X^{(n)})\|^2 | X^{(n)} \right) \leq 2 K_1^2 d \xi_1^2 \xi_2^2 \|X^{(n)} - x^*\|^2 + 2 K_1^2 d^4 \xi_1^4 \xi_2^2 (c^{(n)})^2 \tag{53}$$

Using the recursion (13), we have

$$\begin{aligned} Z^{(n+1)} &:= \|X^{(n+1)} - x^*\|^2 \\ &= \left\| X^{(n)} - x^* + a^{(n)} \left(\frac{f(X^{(n)} + \Delta^{(n)} c^{(n)}) + \varepsilon_1^{(n)} - f(X^{(n)} - \Delta^{(n)} c^{(n)}) - \varepsilon_2^{(n)}}{c^{(n)}} \right) \overline{\Delta^{(n)}} \right\| \\ &= Z^{(n)} + 2 a^{(n)} (X^{(n)} - x^*)^T \left(\widehat{\nabla} f(X^{(n)}) + \frac{\varepsilon^{(n)}}{c^{(n)}} \overline{\Delta^{(n)}} \right) \\ &+ (a^{(n)})^2 \left\| \widehat{\nabla} f(X^{(n)}) + \frac{\varepsilon^{(n)}}{c^{(n)}} \overline{\Delta^{(n)}} \right\|^2 \end{aligned} \tag{54}$$

where $\varepsilon^{(n)} = \varepsilon_1^{(n)} - \varepsilon_2^{(n)}$.

Taking conditional expectations of (54) and using (14) with $\mathbb{E}\varepsilon^{(n)} = 0$, we have

$$\begin{aligned}\mathbb{E}[Z^{(n+1)}|X^{(n)},\Delta^{(n)}] \le{} & Z^{(n)} + 2a^{(n)}(X^{(n)} - x^*)^T\widehat{\nabla} f(X^{(n)}) \\ & + 2(a^{(n)})^2\left\|\widehat{\nabla} f(X^{(n)})\right\|^2 + 2\frac{(a^{(n)})^2}{(c^{(n)})^2}\sigma^2\|\overline{\Delta^{(n)}}\|^2. \end{aligned} \tag{55}$$

If we take expectations (with respect to $\Delta^{(n)}$) of both sides above, using (15) we get

$$\begin{aligned}\mathbb{E}[Z^{(n+1)}|X^{(n)}] \le{} & Z^{(n)} + 2a^{(n)}\mathbb{E}\left[(X^{(n)} - x^*)^T\widehat{\nabla} f(X^{(n)})|X^{(n)}\right] \\ & + 2(a^{(n)})^2\mathbb{E}\left(\|\widehat{\nabla} f(X^{(n)})\|^2|X^{(n)}\right) + 2\xi_2^2\sigma^2\frac{(a^{(n)})^2}{(c^{(n)})^2} \\ \le{} & Z^{(n)} - 2K_0a^{(n)}Z^{(n)} + 2K_1d^2\xi_1^2\xi_2a^{(n)}c^{(n)}\sqrt{Z^{(n)}} \\ & + 4K_1^2d\xi_1^2\xi_2^2(a^{(n)})^2Z^{(n)} + 4K_1^2d^4\xi_1^4\xi_2^2(a^{(n)})^2(c^{(n)})^2 + 2\xi_2^2\sigma^2\frac{(a^{(n)})^2}{(c^{(n)})^2}\end{aligned}$$

where we have used (52) and (53) for the second inequality.

Finally, taking expectations again, using the inequality $\mathbb{E}(\sqrt{Z_n}) \le \sqrt{\mathbb{E}(Z_n)}$, and setting $b_n := \mathbb{E}(Z_n)$, we get the following recursion:

$$\begin{aligned}b_{n+1} \le{} & \left(1 - 2K_0a^{(n)} + 4K_1^2d\xi_1^2\xi_2^2(a^{(n)})^2\right)b_n + 2K_1d^2\xi_1^2\xi_2a^{(n)}c^{(n)}\sqrt{b_n} \\ & + 4K_1^2d^4\xi_1^4\xi_2^2(a^{(n)})^2(c^{(n)})^2 + 2\xi_2^2\sigma^2\frac{(a^{(n)})^2}{(c^{(n)})^2}.\end{aligned} \tag{56}$$

We now note that, the recursion given in (56) is of the same form as inequality (27) in the proof of Theorem 1, so the rest of the proof is step-by-step replication of the proof of Theorem 1.

6. Electronic Companion

An electronic companion to this paper is available as part of the online version that can be found at http://or.journal.informs.org/.

Appendix. Scaled-and-Shifted KW Algorithm

We present a formal statement of the scaled-and-shifted KW algorithm. As in the truncated KW algorithm, we assume knowledge of an interval $I_0 = [l,u]$, that contains the point of maximum. The scaled-and-shifted KW algorithm requires parameter inputs to be set by the user in Step 1. The MATLAB implementation can be downloaded from http://www.columbia.edu/˜mnb2/broadie/research.html/.

Step 1. Setting algorithm parameters and initialization

- h_0: the number of forced hits to boundary points $l + c_n$ and $u - c_n$ by scaling up the $\{a_n\}$ sequence (sample default: 4).
- γ_0: the scaling up factor for the $\{c_n\}$ sequence (sample default: 2).
- k_a: an upper bound on the number of shifts in the $\{a_n\}$ sequence (sample default: 50).
- υ_a: an initial upper bound on the shift in the $\{a_n\}$ sequence (sample default: 10).
- φ_a: an upper bound on the amount of scale up in the $\{a_n\}$ sequence per iteration (sample default: 10).
- k_c: an upper bound on the number of scale-ups in the $\{c_n\}$ sequence (sample default: 50).
- $c^{(0)}$: the parameter defining the maximum possible value of $\{c_n\}$ sequence after the scale-ups; i.e., $c_n \le c^{max} = c^{(0)}(u - l)$ for all $n \ge 1$ (sample default: 0.2).

- m^{max}: an upper bound on the iteration number of the last adaptation in the sequences, i.e., after iteration m^{max} no scaling nor shifting on the sequences is done; we require $m^{max} \geq h_0$ (sample default: total number of iterations).
- g_{max}: the maximum number of gradient estimations allowed to achieve oscillation along each dimension (sample default: 20).
- **Initialization:**

  —X_1: initial starting point; can be random or deterministic but must satisfy $X_1 \in [l+c_1, u-c_1]$.

  —$sh=0$: variable for number of shifts in $\{a_n\}$ sequence.

  —$sc=0$: variable for number of scale-ups in $\{c_n\}$ sequence.

Step 2: The scaling phase Set $n=1$, for $m \leq h_0$,

(a) Calculate X_{n+1} using the recursion given in (1).

(b) Scale up the $\{a_n\}$ sequence, if necessary, ensuring the opposite truncation interval boundary point is hit. Set $g=1$ (counter for number of gradient estimations). While $g \leq g_{max}$

(i) If $X_{n+1} < u-c_n$ and $X_{n+1} > X_n$, find the scale-up factor for $\{a_n\}$ sequence that makes $X_{n+1} = u - c_{n+1}$; i.e., set $\alpha = \min(\varphi_a, (u-c_{n+1}-X_n)/(X_{n+1}-X_n))$ and then use the new sequence $\{a_n\} \leftarrow \{\alpha a_n\}$ for the rest of the iterations. Set $X_{n+1} = u - c_{n+1}$.

(ii) If $X_{n+1} > l + c_n$ and $X_{n+1} < X_n$, find the scale-up factor for $\{a_n\}$ sequence that makes $X_{n+1} = l + c_{n+1}$; i.e., set $\alpha = \min(\varphi_a, (l+c_{n+1}-X_n)/(X_{n+1}-X_n))$ and then use the new sequence $\{a_n\} \leftarrow \{\alpha a_n\}$ for the rest of the iterations. Set $X_{n+1} = l + c_{n+1}$.

(iii) If $n \leq m^{max}$ and $sc \leq \kappa_c$, scale up the $\{c_n\}$ sequence whenever the gradient estimate is too noisy: If $X_{n+1} > X_n = u - c_n$ or $X_{n+1} < X_n = l + c_n$ then calculate $\gamma = \min\{\gamma_0, c^{max}/c_n\}$ and use the new sequence $\{c_n\} \leftarrow \{\gamma c_n\}$ for the rest of the iterations.

(iv) Update $X_{n+1} = \min\{u - c_{n+1}, \max\{X_{n+1}, l + c_{n+1}\}\}$. Increment n and g.

Step 3: The shifting phase Until the termination of the algorithm,

(a) Calculate X_{n+1} using the recursion given in (1).

(b) If $n \leq m^{max}$ and $sh \leq \kappa_a$, shift the $\{a_n\}$ sequence, if necessary, to prevent iterates exiting the truncation interval, but use the upper bound parameter υ_a to prevent a too large shift.

(i) If $X_{n+1} > u - c_{n+1}$, $X_n = l + c_n$ then find the minimum shift in $\{a_n\}$ sequence that makes $X_{n+1} \leq u - c_{n+1}$; i.e.,

- Solve $(u - c_{n+1} - X_n)/\left(\frac{\widetilde{f}(X_n+c_n)-\widetilde{f}(X_n-c_n)}{c_n}\right) = a_{n+\beta'}$ for β'.
- Use $\{a_n\} \leftarrow \{a_{n+\min(\upsilon_a, \lceil \beta' \rceil)}\}$ for the rest of the iterations. If $\min(\upsilon_a, \lceil \beta' \rceil)) = \upsilon_a$, then set $\upsilon_a = 2\upsilon_a$.

(ii) If $X_{n+1} < l + c_{n+1}$, $X_n = u - c_n$ then find the minimum shift in $\{a_n\}$ sequence that makes $X_{n+1} \leq l + c_{n+1}$; i.e.,

- Solve $(l + c_{n+1} - X_n)/\left(\frac{\widetilde{f}(X_n+c_n)-\widetilde{f}(X_n-c_n)}{c_n}\right) = a_{n+\beta'}$ for β'.
- Use $\{a_n\} \leftarrow \{a_{n+\min(\upsilon_a, \lceil \beta' \rceil)}\}$ for the rest of the iterations. If $\min(\upsilon_a, \lceil \beta \rceil)) = \upsilon_a$, then set $\upsilon_a = 2\upsilon_a$.

(c) Repeat Step 2(b)(iii). Set $X_{n+1} = \min\{u - c_{n+1}, \max\{X_{n+1}, l + c_{n+1}\}\}$. Increment n.

Acknowledgments

This work was supported by Credit Suisse and NSF grants DMII-0447652 and DMS-0914539.

References

Andradóttir, S. 1995. A stochastic approximation algorithm with varying bounds. *Operations Research* **43**(6) 1037–1048.

Andradóttir, S. 1996. A scaled stochastic approximation algorithm. *Management Science* **42**(4) 475–498.

Benveniste, A., M. Métivier, P. Priouret. 1990. *Adaptive algorithms and stochastic approximations*. Springer-Verlag, New York Berlin Heidelberg.

Blum, J.R. 1954a. Approximation methods which converge with probability one. *The Annals of Mathematical Statistics* **25**(2) 382–386.

Blum, J.R. 1954b. Multidimensional stochastic approximation methods. *The Annals of Mathematical Statistics* **25**(4) 737–744.

Broadie, M., D.M. Cicek, A. Zeevi. 2010. Multidimensional applications of scaled and shifted stochastic approximation algorithms. Work-in-progress.

Burkholder, D.L. 1956. On a class of stochastic approximation processes. *The Annals of Mathematical Statistics* **27**(4) 1044–1059.

Chen, H. F., T. E. Duncan, B. Pasik-Duncan. 1999. A Kiefer-Wolfowitz algorithm with randomized differences. *IEEE Transactions on Automatic Control* **44**(3) 442–453.

Chung, K.L. 1954. On a stochastic approximation method. *The Annals of Mathematical Statistics* **25**(3) 463–483.

Derman, C. 1956. An application of Chung's lemma to the Kiefer-Wolfowitz stochastic approximation procedure. *The Annals of Mathematical Statistics* **27**(2) 532–536.

Dippon, J. 2003. Accelerated randomized stochastic optimization. *The Annals of Statistics* **31**(4) 1260–1281.

Dippon, J., J. Renz. 1997. Weighted means in stochastic approximation of minima. *SIAM Journal on Control and Optimization* **35**(5) 1811–1827.

Dupac, V. 1957. O Kiefer-Wolfowitzově approximační methodě. *Časopis pro Pěstování Matematiky* **82** 47–75.

Fabian, V. 1967. Stochastic approximation of minima with improved asymptotic speed. *The Annals of Mathematical Statistics* **38**(1) 191–200.

Kesten, H. 1958. Accelerated stochastic approximation. *The Annals of Mathematical Statistics* **29**(1) 41–59.

Kiefer, J., J. Wolfowitz. 1952. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics* **23**(3) 462–466.

Kushner, H.J., G.G. Yin. 2003. *Stochastic approximation and recursive algorithms and applications*. Springer-Verlag, New York.

Lai, T.L. 2003. Stochastic approximation. *Annals of Statistics* **31**(2) 391–406.

Mokkadem, A., M. Pelletier. 2007. A companion for the Kiefer-Wolfowitz-Blum stochastic approximation algorithm. *The Annals of Statistics* **35**(4) 1749–1772.

Nemirovski, A., A. Juditsky, G. Lan, A. Shapiro. 2009. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization* **19**(4) 1574–1609.

Polyak, B.T. 1990. New stochastic approximation type procedures. *Automat. i Telemekh* **7** 98–107.

Polyak, B.T., A.B. Juditsky. 1992. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization* **30**(4) 838–855.

Robbins, H., S. Monro. 1951. A stochastic approximation method. *The Annals of Mathematical Statistics* **22**(3) 400–407.

Ruppert, D. 1988. Efficient estimators from a slowly convergent Robbins-Monro process. Tech. Rep. 781, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York.

Sacks, J. 1958. Asymptotic distribution of stochastic approximation procedures. *The Annals of Mathematical Statistics* **29**(2) 373–405.

Spall, J. C. 1992. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control* **37**(3) 332–341.

Tsybakov, A., B.T. Polyak. 1990. Optimal order of accuracy of search algorithms in stochastic optimization. *Problemy Peredachi Informatsii* **26**(2) 45–53.

Vaidya, R., S. Bhatnagar. 2006. Robust optimization of random early detection. *Telecommunication Systems* **33**(4) 291–316.

Wasan, M.T. 1969. *Stochastic approximation*. Cambridge University Press, London.

This page is intentionally blank. Proper e-companion title page, with INFORMS branding and exact metadata of the main paper, will be produced by the INFORMS office when the issue is being assembled.

Electronic Companion: Multidimensional Extensions

In this electronic companion, we use the new proof technique introduced in the main body of the paper to establish bounds on the mean-squared-error of multidimensional variants of RM and KW algorithms.

EC.1. Bounds on the Multidimensional Robbins-Monro Algorithm

We consider the multidimensional RM algorithm introduced by Blum (1954b). The algorithm is structurized along the following recursion:

$$X^{(n+1)} = X^{(n)} - a^{(n)}\widetilde{g}(X^{(n)}) \tag{EC.1}$$

Here $\widetilde{g}(\cdot)$ is a noisy observation of the function $g(\cdot): \mathbb{R}^d \to \mathbb{R}^d$ and $\{a^{(n)}\}$ is a finite positive real number sequence. We establish bounds on the MSE, $\mathbb{E}\|X^{(n+1)} - x^*\|^2$, of this algorithm where $g(x^*) = 0$. These bounds are formulated in terms of the $\{a^{(n)}\}$ sequence as in Theorem 1. For the underlying function, as in Benveniste et al. (1990), we assume
(G1) $(x - x^*)^T g(x) \geq K_0\|x - x^*\|^2$
(G2) $\mathbb{E}\|\widetilde{g}(x)\|^2 \leq K_1(1 + \|x - x^*\|^2)$
for all $x \in \mathbb{R}^d$, for some finite positive constant $K_0 < K_1$.

The assumptions found in most of the literature on the $\{a^{(n)}\}$ sequence are relaxed with our proof technique. Instead of assuming $\sum_{n=1}^{\infty} a^{(n)} = \infty$ and $\sum_{n=1}^{\infty}(a^{(n)})^2 < \infty$, we assume

$$a^{(n)} \to 0 \text{ as } n \to \infty \text{ and } a^{(n)} \leq a^{(n+1)}(1 + Aa^{(n+1)}) \text{ for all } n \geq 1 \text{ where } A < 2K_0. \tag{EC.2}$$

This assumption still requires $\sum_{n=1}^{\infty} a^{(n)} = \infty$, but does not require $\sum_{n=1}^{\infty}(a^{(n)})^2 < \infty$.

THEOREM EC.1. *Let $\{X^{(n)}\}$ be generated by the Robbins-Monro stochastic approximation recursion given in (EC.1) using $\{a^{(n)}\}$ that satisfies (EC.2). Then under assumptions (G1) and (G2),*

$$\mathbb{E}\|X^{(n+1)} - x^*\|^2 \leq \breve{C}a^{(n)}$$

for all $n \geq 1$, where $\breve{C}$ is a finite positive constant.

From Therorem EC.1, we deduce that the fastest rate of convergence for the RM algorithm is $O(1/n)$ and is achieved by setting $a^{(n)} = \alpha/(n + \beta)$. Any $\{a^{(n)}\}$ sequence that converges faster violates the assumption given in (EC.2).

Proof of Theorem EC.1 **Step 1:** Using the recursion given in (EC.1) we have

$$\begin{aligned} Z^{(n+1)} &:= \|X^{(n+1)} - x^*\|^2 = \left\|X^{(n)} - x^* - a^{(n)}\widetilde{g}(X^{(n)})\right\|^2 \\ &= \|X^{(n)} - x^*\|^2 - 2a^{(n)}(X^{(n)} - x^*)^T\widetilde{g}(X^{(n)}) + (a^{(n)})^2\left\|\widetilde{g}(X^{(n)})\right\|^2. \end{aligned}$$

Taking expectations of both sides conditioned on X^n and using (G1) and (G2) we get

$$\begin{aligned} \mathbb{E}(Z^{(n+1)}|X^{(n)}) &\leq Z^{(n)} - 2K_0a^{(n)}Z^{(n)} + K_1(a^{(n)})^2(1 + \|X^{(n)} - x^*\|^2) \\ &\leq Z^{(n)}\left(1 - 2K_0a^{(n)} + K_1(a^{(n)})^2\right) + K_1(a^{(n)})^2. \end{aligned}$$

Now taking expectations and denoting $b_n := \mathbb{E}(Z^{(n)})$ we get the following real number recursion

$$b_{n+1} \leq b_n(1 - 2K_0a^{(n)} + K_1(a^{(n)})^2) + K_1(a^{(n)})^2 \tag{EC.3}$$

which can be expressed more compactly as $b_{n+1} \leq b_np_n + q_n$ with $p_n := 1 - 2K_0a^{(n)} + K_1(a^{(n)})^2$ and $q_n := K_1(a^{(n)})^2$.

Step 2: In this step of the proof, we establish a bound for sequences satisfying $b_{n+1} \le b_n p_n + q_n$. Note that since $K_1 > K_0 > 0$, p_n is a quadratic equation in $a^{(n)}$ with positive leading coefficient, and negative discriminant, hence $p_n > 0$.

Solving the recursion $b_{n+1} \le b_n p_n + q_n$, we get that for all n

$$b_n \le b_1 \prod_{i=1}^{n} p_i + \sum_{i=2}^{n-1} q_i \prod_{j=i+1}^{n} p_j + q_n =: B_n \tag{EC.4}$$

which provides a crude upper bound on the MSE at the n^{th} step of the algorithm.

Now define

$$n_0 := \sup\{n \ge 1 : (K_1 - 2AK_0)a^{(n)} + K_1 A a_n^2 \ge 2K_0 - A\} + 1 \tag{EC.5}$$

and set $n_0 = 1$ if $(K_1 - 2AK_0)a^{(n)} + K_1 A a_n^2 < 2K_0 - A$ for all $n \ge 1$. Since $A < 2K_0$ we have $n_0 < \infty$ because $a_n \to 0$ as $n \to \infty$. We note that by (EC.5)

$$\zeta := \inf\{2K_0 - A - (K_1 - 2AK_0)a^{(n)} - K_1 A a_n^2 : n \ge n_0\} > 0. \tag{EC.6}$$

Step 3: We now carry out the induction part of the proof; for a detailed introduction of this step see the proof of Theorem 1 in the main body of the paper. For $n = n_0$ suppose $\breve{C}$ is chosen large enough to ensure $\breve{C} \ge B_{n_0+1}/a^{(n_0)} \ge b_{n_0+1}/a^{(n_0)}$. Now suppose $b_{k+1} \le \breve{C} a^{(k)}$ for all $n_0 \le k \le n-1$. We need to show that $b_{n+1} \le \breve{C} a^{(n)}$. Using (EC.3) and the induction hypothesis we have

$$\begin{aligned} b_{n+1} &\le \breve{C} a^{(n-1)}(1 - 2K_0 a^{(n)} + K_1 (a^{(n)})^2) + K_1 (a^{(n)})^2 \\ &\le \breve{C} a^{(n)}(1 + A a^{(n)}) - 2K_0 \breve{C} (a^{(n)})^2 (1 + A a^{(n)}) + K_1 \breve{C} (a^{(n)})^3 (1 + A a^{(n)}) + K_1 (a^{(n)})^2, \end{aligned}$$

where for the second inequality we have used assumption (EC.2). Rearranging terms we get

$$b_{n+1} \le \breve{C} a^{(n)} + (a^{(n)})^2 \left[\breve{C} \left(A - 2K_0 + (K_1 - 2K_0 A) a^{(n)} + K_1 A (a^{(n)})^2 \right) + K_1 \right], \tag{EC.7}$$

Using (EC.6) gives

$$b_{n+1} \le \breve{C} a^{(n)} + (a^{(n)})^2 (-\breve{C}\zeta + K_1). \tag{EC.8}$$

Putting

$$\breve{C} = \max\left\{ \frac{K_1}{\zeta}, \max_{1 \le n \le n_0} \left\{ \frac{B_{n+1}}{a^{(n)}} \right\} \right\}. \tag{EC.9}$$

we have $b_{n+1} \le \breve{C} a_n$ for all $n \ge 1$ and this completes the proof.

EC.2. Bounds on the Multidimensional Kiefer-Wolfowitz Algorithm

Consider the multidimensional optimization problem: $\max_{x \in \mathbb{R}^d} f(x) = \mathbb{E}[\widetilde{f}(x)]$ where $\widetilde{f}$ is a noisy observation of $f : \mathbb{R}^d \to \mathbb{R}$. In order to solve this problem Blum (1954b) introduced a multidimensional version of the KW algorithm. The algorithm uses a finite difference approximation of the gradient in each direction and satisfies the following recursion:

$$X^{(n+1)} = X^{(n)} + a^{(n)} \frac{\widetilde{g}(X^{(n)})}{c^{(n)}}, \quad n = 1, 2, \ldots \tag{EC.10}$$

Here $\widetilde{g}(X^{(n)}) = (\widetilde{f}(X^{(n)} + c^{(n)} e_1) - \widetilde{f}(X^{(n)}), \ldots, \widetilde{f}(X^{(n)} + c^{(n)} e_d) - \widetilde{f}(X^{(n)}))$, and $\{e_1, \ldots, e_d\}$ is the standard basis in $\mathbb{R}^d$.

In this section, we extend the result in Theorem 1 to the setting above. We use the same proof technique as before to establish upper bounds on the MSE of this algorithm. On the observation noise, we assume

$$\sigma^2 := \sup_{x \in \mathbb{R}^d} \mathrm{Var}[\widetilde{f}(x + c e_i) - \widetilde{f}(x) | x] < \infty \tag{EC.11}$$

for all $x \in \mathbb{R}$ and for any $i = 1, \ldots, d$. The gradient of the underlying function $\nabla f(x)$ is assumed to satisfy the following conditions for all $x \in \mathbb{R}^d$:

(G1') $(x-x^*)^T\nabla f(x) \leq -K_0\|x-x^*\|^2$
(G2') $\|\nabla f(x)\| \leq K_1\|x-x^*\|$
where $K_0 < K_1$ are finite positive constants.

THEOREM EC.2. *Let $\{X^{(n)}\}$ be generated by the recursion (EC.10) using $\{a^{(n)}\}$ and $\{c^{(n)}\}$ satisfying (S1)-(S4) with $A < 2K_0$. Then under assumptions (EC.11), (G1'), (G2'), and for τ_1, τ_2 defined in (S4),*

$$\mathbb{E}\|X^{(n+1)}-x^*\|^2 \leq \begin{cases} \breve{C}_1 a^{(n)}/(c^{(n)})^2 & \text{if} \quad (c^{(n)})^4 \leq \tau_1 a^{(n)} \\ \breve{C}_2 (c^{(n)})^2 & \text{if} \quad (c^{(n)})^4 \geq \tau_2 a^{(n)} \end{cases}$$

for all $n \geq 1$, and some finite positive constants $\breve{C}_1$, $\breve{C}_2$.

Proof of Theorem EC.2 **Step 1:** Fix $\{a^{(n)}\}$ and $\{c^{(n)}\}$ as in the statement of the theorem. Setting $\nabla f_i(x) := \partial f(x)/\partial x_i$ and using Taylor expansion, for $x \in \mathbb{R}^d$ and $c \in \mathbb{R}_+$, there exist $T = [T_1, \ldots, T_d]$ with $0 \leq T_i \leq 1$ for all $i = 1, \ldots, d$, such that $f(x+ce_i) = f(x) + c\nabla f_i(x+T_i ce_i)$, since e_i has all zero entries except its i^{th} coordinate.

Therefore, we have

$$\widehat{\nabla} f_i(x) := \frac{f(x+c^{(n)}e_i)-f(x)}{c^{(n)}} = \nabla f_i(x+T_i c^{(n)} e_i). \tag{EC.12}$$

Defining $\widehat{\nabla} f(x) := (\widehat{\nabla} f_1(x), \ldots, \widehat{\nabla} f_d(x))$ and using (EC.12), we have

$$\begin{aligned}
(x-x^*)^T\widehat{\nabla} f(x) &= (x-x^*)^T\nabla f(x+Tc^{(n)}) \\
&= (x+Tc^{(n)}-x^*)^T\nabla f(x+Tc^{(n)}) - c^{(n)}T^T\nabla f(x+Tc^{(n)}) \\
&\leq -K_0\|x-x^*+Tc^{(n)}\|^2 + K_1 c^{(n)}\|x+Tc^{(n)}-x^*\| \\
&= -K_0\left(\|x-x^*\|^2 + 2c^{(n)}(x-x^*)^T T + \|Tc^{(n)}\|^2\right) + K_1 c^{(n)}\|x+Tc^{(n)}-x^*\| \\
&\leq -K_0\|x-x^*\|^2 + 2K_0 c^{(n)}\|x-x^*\| + K_1 c^{(n)}\|x-x^*\| + K_1 (c^{(n)})^2 \qquad \text{(EC.13)}
\end{aligned}$$

Here, the first inequality follows from assumptions (G1') and (G2') and the last one follows from simple algebra and subadditivity of the Euclidean norm.

Using (EC.12) and (G2'), we obtain

$$\begin{aligned}
\|\widehat{\nabla} f(x)\|^2 &= \|\nabla f(x+Tc^{(n)})\|^2 \\
&\leq K_1^2\|x+Tc^{(n)}-x^*\|^2 \\
&\leq 2K_1^2\|x-x^*\|^2 + 2K_1^2(c^{(n)})^2. \qquad \text{(EC.14)}
\end{aligned}$$

where the last inequality follows from the inequality $\|a+b\|^2 \leq 2(\|a\|^2+\|b\|^2)$.

Step 2: Let $X^{(n)}$ be the output of the n^{th} iterate of (EC.10). Note that

$$\mathbb{E}\left(\frac{\widetilde{f}(X^{(n)}+c^{(n)}e_i)-\widetilde{f}(X^{(n)})}{c^{(n)}} \,\Big|\, X^{(n)}\right) = \widehat{\nabla} f_i(X^{(n)}),$$

which together with bounded variance assumption and (EC.14) implies that

$$\begin{aligned}
\mathbb{E}\left[\left\|\frac{\widetilde{g}(X^{(n)},c^{(n)})}{c^{(n)}}\right\|^2 \,\Big|\, X^{(n)}\right] &= \sum_{i=1}^{d}\text{Var}\left(\frac{f(X^{(n)}+c^{(n)}e_i)-\widetilde{f}(X^{(n)})}{c^{(n)}} \,\Big|\, X^{(n)}\right) + \|\widehat{\nabla} f(X^{(n)})\|^2 \\
&\leq \frac{d\sigma^2}{(c^{(n)})^2} + 2K_1^2\|x-x^*\|^2 + 2K_1^2(c^{(n)})^2. \qquad \text{(EC.15)}
\end{aligned}$$

Now, using (EC.10) we have

$$\begin{aligned} Z^{(n+1)} &:= \|X^{(n+1)} - x^*\|^2 = \left\| X^{(n)} - x^* + a^{(n)} \frac{\widetilde{g}(X^{(n)}, c^{(n)})}{c^{(n)}} \right\|^2 \\ &= \|X^{(n)} - x^*\|^2 + 2a^{(n)}(X^{(n)} - x^*)^T \frac{\widetilde{g}(X^{(n)}, c^{(n)})}{c^{(n)}} + (a^{(n)})^2 \left\| \frac{\widetilde{g}(X^{(n)}, c^{(n)})}{c^{(n)}} \right\|^2 . \end{aligned}$$

Taking expectations of both sides conditioned on $X^{(n)}$ and using (EC.15) we get

$$\mathbb{E}(Z^{(n+1)}|X^{(n)}) \leq Z^{(n)} + 2a^{(n)}(X^{(n)} - x^*)^T \widehat{\nabla} f(X^{(n)}) + (a^{(n)})^2 \left(\frac{d\sigma^2}{(c^{(n)})^2} + 2K_1^2 Z^{(n)} + 2K_1^2 (c^{(n)})^2 \right).$$

Using the bound in (EC.13) we have

$$\begin{aligned} \mathbb{E}(Z^{(n+1)}|X^{(n)}) &\leq Z^{(n)} - 2a^{(n)}K_0 Z^{(n)} + 2(2K_0 + K_1)a^{(n)}c^{(n)}\sqrt{Z^{(n)}} + 2K_1 a^{(n)}(c^{(n)})^2 \\ &+ (a^{(n)})^2 \left(\frac{d\sigma^2}{(c^{(n)})^2} + 2K_1^2 Z^{(n)} + 2K_1^2 (c^{(n)})^2 \right). \end{aligned}$$

Finally, taking expectations, using Jensen's inequality, and setting $b_n := \mathbb{E}(Z^{(n)})$ we get the following recursion:

$$\begin{aligned} b_{n+1} &\leq \left(1 - 2a^{(n)}K_0 + 2K_1^2(a^{(n)})^2\right) b_n + (4K_0 + 2K_1)a^{(n)}c^{(n)}\sqrt{b_n} \\ &+ 2K_1 a^{(n)}(c^{(n)})^2 + (a^{(n)})^2 \left(\frac{d\sigma^2}{(c^{(n)})^2} + 2K_1^2 (c^{(n)})^2 \right). \end{aligned} \tag{EC.16}$$

Step 3: Before we start the induction proof, we will derive a crude upper bound on b_n that will be used later. Using $\sqrt{b_n} \leq 1 + b_n$ in (EC.16) we get

$$\begin{aligned} b_{n+1} &\leq \left(1 - 2a^{(n)}K_0 + 2K_1^2(a^{(n)})^2 + (4K_0 + 2K_1)a^{(n)}c^{(n)}\right) b_n \\ &+ (4K_0 + 2K_1)a^{(n)}c^{(n)} + 2K_1 a^{(n)}(c^{(n)})^2 + (a^{(n)})^2 \left(\frac{d\sigma^2}{(c^{(n)})^2} + 2K_1^2 (c^{(n)})^2 \right), \end{aligned}$$

which can be expressed more compactly as $b_{n+1} \leq b_n p_n + q_n$ with

$$\begin{aligned} p_n &:= 1 - 2a^{(n)}K_0 + 2K_1^2(a^{(n)})^2 + (4K_0 + 2K_1)a^{(n)}c^{(n)} > 0, \\ q_n &:= (4K_0 + 2K_1)a^{(n)}c^{(n)} + 2K_1 a^{(n)}(c^{(n)})^2 + (a^{(n)})^2 \left(\frac{d\sigma^2}{(c^{(n)})^2} + 2K_1^2 (c^{(n)})^2 \right). \end{aligned}$$

Note that since $(4K_0 + 2K_1)a^{(n)}c^{(n)} > 0$, we have $p_n \geq 1 - 2a^{(n)}K_0 + 2K_1^2(a^{(n)})^2$ which is a quadratic equation in $a^{(n)}$ with positive leading coefficient, and $0 < K_0 < K_1$ ensures it has negative discriminant, hence $p_n > 0$. As in the proof of Theorem EC.1, we get $b_n \leq B_n$ with the new definitions of p_n and q_n, where B_n is defined in (EC.4).

Put

$$n_0' := \sup\{n \geq 1 : (2K_1^2 - 2K_0 A)a^{(n)} + 2K_1^2 A(a^{(n)})^2 \geq 2K_0 - A\} + 1, \tag{EC.17}$$

and set $n_0' = 1$ if $(2K_1^2 - 2K_0 A)a^{(n)} + 2K_1^2 A(a^{(n)})^2 < 2K_0 - A$ for all n. Since we assume $A < 2K_0$, we have $n_0' < \infty$ because $a^{(n)} \to 0$ as $n \to \infty$ (assumption (S3')). Also, note that by (EC.17)

$$\zeta := \inf\{2K_0 - A - (2K_1^2 - 2K_0 A)a^{(n)} + 2K_1^2 A(a^{(n)})^2 : n \geq n_0'\} > 0. \tag{EC.18}$$

Step 4: Now we carry out the induction part of the proof.

Case (i): Suppose $(c^{(n)})^4/a^{(n)} \leq \tau_1$, for all $n \geq 1$. We will first show that $b_{n+1} \leq \breve{C}_1 a^{(n)}/(c^{(n)})^2$ for all $n \geq n_0'$ and some finite positive constant $\breve{C}_1$. First, for $n = n_0'$ suppose $\breve{C}_1$ is chosen large enough to ensure $\breve{C}_1 \geq B_{n_0'+1}(c^{(n_0')})^2/a^{(n_0')} \geq b_{n_0'+1}(c^{(n_0')})^2/a^{(n_0')}$. Now fix $n > n_0'$ and suppose $b_{k+1} \leq \breve{C}_1 a^{(k)}/(c^{(k)})^2$ for all $n_0' \leq k \leq n-1$. We need to show that $b_{n+1} \leq \breve{C}_1 a^{(n)}/(c^{(n)})^2$. Using inequality (EC.16) and the induction hypothesis we have

$$\begin{aligned} b_{n+1} \leq\; & \left(1 - 2a^{(n)}K_0 + 2K_1^2(a^{(n)})^2\right)\breve{C}_1 \frac{a^{(n-1)}}{(c^{(n-1)})^2} + (4K_0 + 2K_1)a^{(n)}c^{(n)}\sqrt{\breve{C}_1}\frac{\sqrt{a^{(n-1)}}}{c^{(n-1)}} \\ & + 2K_1 a^{(n)}(c^{(n)})^2 + (a^{(n)})^2\left(\frac{d\sigma^2}{(c^{(n)})^2} + 2K_1^2(c^{(n)})^2\right) \\ \leq\; & \breve{C}_1\frac{a^{(n)}}{(c^{(n)})^2}(1 + Aa^{(n)}) - 2K_0\breve{C}_1\frac{(a^{(n)})^2}{(c^{(n)})^2}(1 + Aa^{(n)}) + 2K_1^2\breve{C}_1\frac{(a^{(n)})^3}{(c^{(n)})^2}(1 + Aa^{(n)}) \\ & + (4K_0 + 2K_1)(a^{(n)})^{3/2}\sqrt{\breve{C}_1}\left(1 + \frac{A}{2}a^{(n)}\right) + 2K_1 a^{(n)}(c^{(n)})^2 + (a^{(n)})^2\left(\frac{d\sigma^2}{(c^{(n)})^2} + 2K_1^2(c^{(n)})^2\right), \end{aligned}$$

where for the second inequality we have used condition (S1) and the inequality $\sqrt{1 + Aa_n} \leq 1 + Aa_n/2$. Rearranging terms we get

$$\begin{aligned} b_{n+1} \leq\; & \breve{C}_1\frac{a^{(n)}}{(c^{(n)})^2} + \frac{(a^{(n)})^2}{(c^{(n)})^2}\Big\{\breve{C}_1[A - 2K_0 + (2K_1^2 - 2K_0A)a^{(n)} + 2K_1^2A(a^{(n)})^2] \\ & + (4K_0 + 2K_1)\sqrt{\breve{C}_1}\left[\frac{(c^{(n)})^2}{\sqrt{a^{(n)}}} + \frac{A}{2}\sqrt{a^{(n)}}(c^{(n)})^2\right] + 2K_1\frac{(c^{(n)})^4}{a^{(n)}} + d\sigma^2 + 2K_1^2(c^{(n)})^4 \;\Big\}. \end{aligned}$$

Let ν and κ denote the upper bounds on the $\{a^{(n)}\}$ and $\{c^{(n)}\}$ sequences, respectively. Using $(c^{(n)})^4/a^{(n)} \leq \tau_1$ with (EC.18) gives:

$$b_{n+1} \leq \breve{C}_1\frac{a^{(n)}}{(c^{(n)})^2} + \frac{(a^{(n)})^2}{(c^{(n)})^2}\left[-\breve{C}_1\zeta + (4K_0 + 2K_1)\sqrt{\breve{C}_1}(\sqrt{\tau_1} + \frac{A}{2}\sqrt{\nu}\kappa^2) + 2K_1\tau_1 + d\sigma^2 + 2K_1^2\kappa^4\right]. \tag{EC.19}$$

Now, if we can show that for some finite positive constant $\breve{C}_1$,

$$-\zeta\breve{C}_1 + (4K_0 + 2K_1)(\sqrt{\tau_1} + \frac{A}{2}\sqrt{\nu}\kappa^2)\sqrt{\breve{C}_1} + 2K_1\tau_1 + d\sigma^2 + 2K_1^2\kappa^4 \leq 0, \tag{EC.20}$$

then the induction proof would be complete. Viewing this as a quadratic in $\sqrt{\breve{C}_1}$, we first observe that the leading coefficient is negative, by (EC.18). It follows that this quadratic admits a solution, in particular, solving for the positive root and using $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$, we have $b_{n+1} \leq \breve{C}_1 a^{(n)}/(c^{(n)})^2$ for all $n \geq n_0'$ with any choice of $\breve{C}_1$ satisfying

$$\breve{C}_1 \geq \max\left\{\left[\frac{(4K_0 + 2K_1)(\sqrt{\tau_1} + \frac{A}{2}\sqrt{\nu}\kappa^2)}{\zeta} + \sqrt{\frac{2K_1\tau_1 + d\sigma^2 + 2K_1^2\kappa^4}{\zeta}}\right]^2, \frac{(c^{(n_0')})^2}{a^{(n_0')}}B_{n_0'+1}\right\}. \tag{EC.21}$$

Finally let us modify the constant $\breve{C}_1$ so that the result holds for all $n \geq 1$. This requires a simple modification in (EC.21). In particular, using the fact that $b_n \leq B_n$ it can be done by setting

$$\breve{C}_1 = \max\left\{\left[\frac{(4K_0 + 2K_1)(\sqrt{\tau_1} + \frac{A}{2}\sqrt{\nu}\kappa^2)}{\zeta} + \sqrt{\frac{2K_1\tau_1 + d\sigma^2 + 2K_1^2\kappa^4}{\zeta}}\right]^2, \max_{1 \leq n \leq n_0'}\left\{\frac{(c^{(n)})^2}{a^{(n)}}B_{n+1}\right\}\right\}. \tag{EC.22}$$

Case (ii): Suppose $(c^{(n)})^4/a^{(n)} \geq \tau_2$, for all $n \geq 1$. Using similar steps to those in the proof of case (i), we will first show that $b_{n+1} \leq \breve{C}_2(c^{(n)})^2$ for all $n \geq n_0'$ for some finite positive constant $\breve{C}_2$. First, for $n = n_0'$ suppose $\breve{C}_2$ is chosen large enough to assure $\breve{C}_2 \geq B_{n_0'+1}/(c^{(n_0')})^2 \geq b_{n_0'+1}/(c^{(n_0')})^2$. Now suppose we have $b_{k+1} \leq \breve{C}_2(c^{(k)})^2$ for all $n_0' \leq k \leq n-1$. We need to prove $b_{n+1} \leq \breve{C}_2(c^{(n)})^2$. Using inequality (EC.16) and the induction hypothesis we have

$$\begin{aligned}
b_{n+1} \leq{} & \left(1 - 2a^{(n)}K_0 + 2K_1^2(a^{(n)})^2\right)\breve{C}_2(c^{(n-1)})^2 + (4K_0+2K_1)a^{(n)}c^{(n)}\sqrt{\breve{C}_2}c^{(n-1)} \\
& + 2K_1a^{(n)}(c^{(n)})^2 + (a^{(n)})^2\left(\frac{d\sigma^2}{(c^{(n)})^2} + 2K_1^2(c^{(n)})^2\right) \\
\leq{} & \breve{C}_2(c^{(n)})^2(1+Aa^{(n)}) - 2K_0\breve{C}_2a^{(n)}(c^{(n)})^2(1+Aa^{(n)}) + 2K_1^2\breve{C}_2(a^{(n)})^2(c^{(n)})^2(1+Aa^{(n)}) \\
& + (4K_0+2K_1)a^{(n)}(c^{(n)})^2\sqrt{\breve{C}_2}\left(1+\frac{A}{2}a^{(n)}\right) + 2K_1a^{(n)}(c^{(n)})^2 + (a^{(n)})^2\left(\frac{d\sigma^2}{(c^{(n)})^2} + 2K_1^2(c^{(n)})^2\right),
\end{aligned}$$

where for the second inequality we have used (S2) with the inequality $\sqrt{1+Aa^{(n)}} \leq 1 + Aa^{(n)}/2$. Rearranging terms we get

$$\begin{aligned}
b_{n+1} \leq{} & \breve{C}_2(c^{(n)})^2 + a^{(n)}(c^{(n)})^2\Big\{\breve{C}_2[A - 2K_0 + (2K_1^2 - 2K_0A)a^{(n)} + 2K_1^2A(a^{(n)})^2] \\
& + \sqrt{\breve{C}_2}(4K_0+2K_1)\left(1+\frac{A}{2}a^{(n)}\right) + 2K_1 + d\sigma^2\frac{a^{(n)}}{(c^{(n)})^4} + 2K_1^2a^{(n)}\Big\}.
\end{aligned}$$

Using $a^{(n)} \leq \nu$, (EC.18) and the assumption that $a^{(n)}/(c^{(n)})^4 \leq 1/\tau_2$, we get

$$b_{n+1} \leq \breve{C}_2(c^{(n)})^2 + a^{(n)}(c^{(n)})^2\left[-\breve{C}_2\zeta + \sqrt{\breve{C}_2}(4K_0+2K_1)\left(1+\frac{A\nu}{2}\right) + 2K_1 + \frac{d\sigma^2}{\tau_2} + 2K_1^2\nu\right]. \tag{EC.23}$$

Similar to the first case, we need

$$-\breve{C}_2\zeta + \sqrt{\breve{C}_2}(4K_0+2K_1)\left(1+\frac{A\nu}{2}\right) + 2K_1 + \frac{d\sigma^2}{\tau_2} + 2K_1^2\nu \leq 0,$$

for a suitable choice of $\breve{C}_2$. Using the same argument as before, we have $b_{n+1} \leq \breve{C}_2(c^{(n)})^2$ for all $n \geq n_0'$ with any $\breve{C}_2$ satisfying

$$\breve{C}_2 \geq \max\left\{\left[\frac{(4K_0+2K_1)\left(1+\frac{A\nu}{2}\right)}{\zeta} + \sqrt{\frac{2K_1 + \frac{d\sigma^2}{\tau_2} + 2K_1^2\nu}{\zeta}}\right]^2, \frac{1}{(c^{(n_0')})^2}B_{n_0'+1}\right\}. \tag{EC.24}$$

Setting

$$\breve{C}_2 = \max\left\{\left[\frac{(4K_0+2K_1)\left(1+\frac{A\nu}{2}\right)}{\zeta} + \sqrt{\frac{2K_1 + \frac{d\sigma^2}{\tau_2} + 2K_1^2\nu}{\zeta}}\right]^2, \max_{1\leq n\leq n_0'}\left\{\frac{1}{(c^{(n)})^2}B_{n+1}\right\}\right\}. \tag{EC.25}$$

we get the result for all $n \geq 1$ and this completes the proof