Mathematical Programming
https://doi.org/10.1007/s10107-021-01667-6

FULL LENGTH PAPER

Series B

Adaptive Sampling line search for local stochastic optimization with integer variables

Prasanna K. Ragavan[1] · Susan R. Hunter[2] · Raghu Pasupathy[3] · Michael R. Taaffe[4]

Received: 2 April 2020 / Accepted: 13 May 2021
© Springer-Verlag GmbH Germany, part of Springer Nature and Mathematical Optimization Society 2021

Abstract

We consider optimization problems with an objective function that is estimable using a Monte Carlo oracle, constraint functions that are known deterministically through a constraint-satisfaction oracle, and integer decision variables. Seeking an appropriately defined local minimum, we propose an iterative adaptive sampling algorithm that, during each iteration, performs a statistical local optimality test followed by a line search when the test detects a stochastic descent direction. We prove a number of results. First, the true function values at the iterates generated by the algorithm form an almost-supermartingale process, and the iterates are absorbed with probability one into the set of local minima in finite time. Second, such absorption happens exponentially fast in iteration number and in oracle calls. This result is analogous to non-standard rate guarantees in stochastic continuous optimization contexts that involve sharp minima. Third, the oracle complexity of the proposed algorithm increases linearly in the dimensionality of the local neighborhood. As a solver, primarily due

S. R. Hunter thanks the National Science Foundation for support under grant CMMI-1554144. R. Pasupathy thanks the Office of Naval Research for support provided by the grants N000141712295 and 13000991.

✉ Raghu Pasupathy
pasupath@purdue.edu

Prasanna K. Ragavan
pracha87@gmail.com

Susan R. Hunter
susanhunter@purdue.edu

Michael R. Taaffe
taaffe@vt.edu

1 Llamasoft, Inc., Ann Arbor, MI, USA

2 School of Industrial Engineering, Purdue University, West Lafayette, IN, USA

3 Department of Statistics, Purdue University, West Lafayette, IN, USA

4 Virginia Tech, Blacksburg, VA, USA

Published online: 09 July 2021

to combining line searches that use common random numbers with statistical tests for local optimality, the proposed algorithm is effective on a variety of problems. We illustrate such performance using three problem suites, on problems ranging from 25 to 200 dimensions.

Keywords Local stochastic optimization · Simulation optimization · Adaptive sampling · Integer variables

Mathematics Subject Classification 49Mxx · 65Kxx · 90Bxx · 90Cxx

1 Introduction

We consider simulation optimization (SO) problems in which the objective function is an expectation that can only be observed with stochastic error, and the decision variables can only assume integer values. Formally, we state the SO problem with integer decision variables as

$$\begin{aligned} &\text{minimize:} \quad f(x) := \mathbb{E}[F(x,\xi)] = \int F(x,\xi) dP(\xi) \\ &\text{subject to:} \quad g_i(x) \le 0, i = 1,2,\ldots,r; \quad x \in \mathbb{Z}^d, \end{aligned} \tag{Q}$$

where $f : \mathcal{D} \to \mathbb{R}$, $\mathcal{D} \subseteq \mathbb{R}^d$ is the real-valued objective function, P is the probability measure induced by the random variable ξ, the constraint functions $g_i : \mathcal{D} \to \mathbb{R}$, $i = 1,2,\ldots,r$ are real-valued, and $\mathbb{Z}^d$ is the d-dimensional integer lattice. We allow the constraints to be *hidden* [26]; that is, they may be observed as the output of a deterministic constraint-satisfaction oracle. Thus, the oracle reveals only whether the requested x satisfies the constraint set $g_i(x) \le 0, i = 1,2,\ldots,r$. Accordingly, we define the feasible set as

$$\mathcal{X} := \{x \in \mathcal{D} \cap \mathbb{Z}^d : g_i(x) \le 0, i = 1,2,\ldots,r\},$$

which may be very large or countably infinite.

Further, we consider the context of solving Problem Q only to local optimality. Formally, we say that a feasible point $x^* \in \mathcal{X}$ is an $\mathcal{N}_1$-*local solution* to Problem Q if $f(x^*) \le f(x)$ for all $x \in \mathcal{N}_1(x^*) \cap \mathcal{X}$, where

$$\mathcal{N}_1(x) := \{x \in \mathbb{Z}^d : \|x - x^*\| \le 1\}$$

and $\|\cdot\|$ refers to the L_2 norm. The set of all $\mathcal{N}_1$-local solutions is $\mathcal{L}(\mathcal{N}_1)$. This context enables us to exploit local structure when finding a point in $\mathcal{L}(\mathcal{N}_1)$; algorithms developed in this context may later be incorporated into a method that seeks a global solution.

In solving Problem Q to local optimality, we assume that the objective function f can only be observed with error at each feasible point in the integer lattice. Let

$\varepsilon_i, i = 1, 2, \ldots, m$ be *random fields* that specify the error

$$\varepsilon_i(\cdot) \coloneqq F(\cdot, \xi_i) - f(\cdot), \tag{1}$$

where $\xi_i, i = 1, 2, \ldots, m$ are independent and identically distributed (i.i.d.) random variables. Then, the objective function estimator is

$$\bar{F}(\cdot, m) \coloneqq f(\cdot) + m^{-1} \sum_{i=1}^{m} \varepsilon_i(\cdot). \tag{2}$$

We emphasize that $\bar{F}(\cdot, m)$ is observed *pointwise* in $\mathcal{X}$, that is, the oracle takes as input the decision variable $x \in \mathcal{X}$ and the random variables $\xi_i, i = 1, 2, \ldots, m$, and outputs the objective function estimate

$$\bar{F}(x, m) = m^{-1} \sum_{i=1}^{m} F(x, \xi_i)$$

at the decision variable x. Estimating the objective function at a specific point $x \in \mathcal{X}$ thus involves obtaining m function observations (or simulation replications) from a Monte Carlo simulation oracle or *drawing* m observations from a given database of scenarios, and then constructing the estimator as the sample mean of the obtained observations. And, in particular, the entire field $\bar{F}(\cdot, m)$ is not observed at once. In this sense, the problem we consider is the stochastic analogue of deterministic black box oracle optimization [8,34]. (As the feasible region is a subset of $\mathbb{Z}^d$, the oracle is necessarily a zeroth-order stochastic black box oracle.) Henceforth, we refer to the Monte Carlo simulation oracle or database as the *stochastic oracle*.

SO problems of the nature we describe, called integer-ordered SO problems by [13,38,39], constitute an important problem class. These problems arise in a variety of applications including, but not limited to, inventory replenishment policy optimization [17], revenue management [11], planning and scheduling [24,28], and determining vaccine allocation strategies to prevent an epidemic outbreak [10]. At the time of writing, about 40% of the problems in the `simopt` problem library [13] are integer-ordered SO problems.

As a practical matter, obtaining a single observation of the objective function value at x, $F(x, \cdot)$, may require significant computational resources — on the order of several seconds, minutes, or hours, depending on the application. Given the computational expense, naïve solution methods, such as using the same large sample size at every point visited by an algorithm, are unlikely to be useful. Instead, these problems require sophisticated algorithms that appropriately control stochastic error and return a high-quality solution with as few function evaluations as possible. Any reduction in the number of function evaluations required to find a high-quality solution is likely to enable decision-makers to solve important problems faster, or to solve new problems they previously considered unsolvable.

1.1 Our approach and key results

Toward solving Problem Q, we propose an *adaptive sampling* [12,40,42,46] algorithm called ADALINE that repeatedly performs a statistical local optimality test followed

by a line search executed along a stochastic descent direction. The statistical local optimality test consists of adaptively sampling points in a neighborhood of the current candidate iterate until it finds a statistically better neighbor; the sample size required to identify the statistically better neighbor is subsequently used in the procedures that conduct the line search and identify new stochastic descent directions. Thus, ADALINE is adaptive in the strict sense, that is, the sample size used for function estimation within an iteration is a *stopping time* with respect to the sigma algebra of generated observations.

We demonstrate that ADALINE's iterates exhibit a certain strong type of almost sure convergence. First, we prove that the true function values at the iterates generated by the algorithm "almost" form a supermartingale [51, p. 94], which we discuss further in Sect. 2, and are *absorbed* with probability one into the set of local minima after a finite number of stochastic oracle calls (Theorem 1). Furthermore, under a mild lower bound on the stopping times corresponding to the expended sample sizes in the stochastic optimality test, we show that absorption into the set of local minima occurs (Theorem 2) exponentially fast in the iteration number (Theorem 3) and the number of oracle calls (Theorem 4). The last result (Theorem 4) stands in contrast to stochastic continuous optimization where the work complexity guarantee is usually much weaker [3], except in contexts involving sharp minima [22]. We also demonstrate that the oracle complexity increases linearly in the dimensionality of the problem (Theorem 4).

The theoretical results describing ADALINE's behavior indicate a particularly strong form of convergence and fast convergence rate properties. Along with a few other stipulations, the results rely heavily on two key properties:

(a) If the current iterate is not an $\mathcal{N}_1$-local solution, the statistical local optimality test terminates in finite time almost surely (Lemma 5); likewise, if the point being tested is optimal, there is positive probability that the test never terminates.
(b) After each statistical local optimality test, the next line search direction is the max-descent (analogue of the negative gradient) direction with positive probability (Property 1).

Although they are important requirements for convergence, properties like (a) and (b) alone do not fully capture the requirements needed for ADALINE to perform efficiently as a solver, especially in high dimensions. To this end, several heuristics embedded in ADALINE ensure efficiency in the statistical local optimality test and in finding a "good" stochastic descent direction:

(c) In the statistical local optimality test at the incumbent iterate, ADALINE uses a discrete probability measure to select a feasible neighbor from which to obtain additional objective function observations. ADALINE frequently updates the discrete probability measure using weights that resemble the Student's t statistic [9,18,41,47] to ensure a higher probability of sampling points deemed likely to be better than the incumbent.
(d) When finding a new stochastic descent direction at the end of a line search, ADALINE first constructs an estimated descent cone from the feasible neighborhood points that are estimated as better than the incumbent iterate; the descent cone may be a small sub-space of descending feasible directions when operating close to a

constraint. Then, ADALINE constructs an estimated descent direction by appropriately weighting all directions in the estimated descent cone. The chosen weights are standardized estimated optimality gaps that resemble the Student's t statistic.

The choices (c) and (d) are heuristic in the sense that they never directly enter the convergence rate calculations. However, they are crucial for efficient implementation in high dimensions.

Finally, while we have stated Problem Q as seeking an $\mathcal{N}_1$-local solution, our results extend to problems that seek an $\mathcal{N}_a$-local solution in an analogous manner, where a feasible point $x^* \in \mathcal{X}$ is an $\mathcal{N}_a$-*local solution* to Problem Q if $f(x^*) \leq f(x)$ for all $x \in \mathcal{N}_a(x^*) \cap \mathcal{X}$, and

$$\mathcal{N}_a(x) \coloneqq \{x \in \mathbb{Z}^d : \|x - x^*\| \leq a\}, \quad a \geq 1. \tag{3}$$

Such extension to larger neighborhoods is seamless, at least in principle, because of our proposed algorithm's modular structure, where only the statistical local optimality test is affected by the chosen value of a in (3). Furthermore, our statistical local optimality test is a ranking and selection procedure [16,27,41] that can be executed on sets of the type $\mathcal{N}_a(x)$. As a comment on further generalization, we observe that our assumption of the feasible set $\mathcal{X}$ being a subset of the d-dimensional integers $\mathbb{Z}^d$ is primarily for easy exposition. Our results extend to countable metric spaces $\mathcal{X}$ that have no accumulation points, thereby subsuming a vast array of problem contexts arising in graph networks and genomics [10,31] that have recently become fertile.

1.2 Existing methods

Black-box oracle methods for solving constrained integer and mixed-integer problems have a long history, resulting in a vast and mature literature. See [1,2,4,6,29,30] and references therein for an opening into this area. The corresponding literature in the stochastic context is sparse, especially when seeking a local solution for Problem Q, is relatively much smaller. See [15,44,48] for recent work on methods that guarantee convergence to a global solution, and [53,54] for those that exploit convexity in discrete spaces [33]. As our focus is on methods that converge to a local solution, we expand only upon this literature, including the popular partitioning-based algorithms COMPASS [14] and Industrial Strength COMPASS (ISC) [52], as well as the pseudogradient-based line search algorithm R- SPLINE [50].

First, operating on an explicitly-defined feasible space, COMPASS iteratively partitions and expends simulation effort to evaluate feasible points from the "most promising area" of the feasible space to guarantee convergence to a local solution. At the end of every iteration, COMPASS updates the most promising area by identifying and grouping solutions that are "close" to the best solution and then solving an optimization problem. COMPASS implicitly exploits the structure of the objective function, although the requirement to solve an optimization problem affects scalability to higher dimensions. The Industrial Strength COMPASS (ISC), an enhancement over COMPASS, returns good quality local solutions by restarting COMPASS several

times from promising initial points obtained using a genetic algorithm. Then, a ranking and selection [21] algorithm identifies the best of the local solutions.

Perhaps the closest algorithm to the present work, R- SPLINE, is a line search based algorithm that identifies a local minimum by operating within a retrospective approximation (RA) framework [37]. An RA framework prescribes solving a sequence of sample-path problems with increasing sample sizes; each sample-path problem solved constitutes one RA iteration. Within an RA iteration of R- SPLINE, the SPLINE sub-routine repeatedly conducts a pseudo-gradient based line search followed by a neighborhood enumeration test for local optimality. Upon locating a confirmed sample-path local minimum, the current RA iteration ends, and the next RA iteration begins by using the solution from the last RA iteration as a warm start.

It is worthwhile to observe that while R- SPLINE is designed to control stochastic error, it uses a pre-specified, deterministic rule for determining the sample size sequence, possibly leading to inefficiencies by preventing a reversion to smaller sample sizes as the algorithm progresses. In a sense, the algorithm proposed in this paper remedies this issue; it combines arguably the best aspect of R- SPLINE, line search, with a module that adaptively decides how much to sample based on measures of proximity to a solution. We compare a version of ADALINE with the implementation of R- SPLINE from the `simopt.org` solver library in Sect. 5.

2 Preliminaries

We discuss notation, definitions, useful results, and standing assumptions.

2.1 Notation, definitions, and useful Results

First, we define notation. For $x \in \mathbb{R}$, $x^+ := \max\{0, x\}$. For $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$, $\|x\| := (\sum_{j=1}^d x_j^2)^{1/2}$ is the Euclidean norm. For a sequence of real numbers (a_n), we say $a_n = O(1)$ if there exists $c \in (0, \infty)$ such that $|a_n| < c$ for large enough n. For a sequence of events $(\mathcal{A}_n)_{n\geq 0}$ defined on a probability space, we say that $\mathcal{A}_n$ *i.o.* ("infinitely often") if infinitely many of $\mathcal{A}_n$ occur, where $\mathcal{A}_n$ *i.o.* $:= \limsup_n \mathcal{A}_n = \cap_{n=1}^{\infty} \cup_{j=n}^{\infty} \mathcal{A}_j$. For an event $\mathcal{A}$, $\mathcal{A}$ *with probability one (w.p.1)* means $P\{\mathcal{A}\} = 1$.

The following definition of a deleted neighborhood is useful throughout the paper to denote the neighborhood of a point while excluding the point itself.

Definition 1 The *deleted neighborhood* of $x \in \mathcal{X}$ is $\mathcal{N}_1'(x) := \mathcal{N}_1(x) \setminus \{x\}$.

In the convergence proofs, we require the notions of a max-descent function and a mildly-coercive function. The max-descent of f at x represents the magnitude of steepest descent of f at x.

Definition 2 The max-descent function of f, denoted $f_\Delta^- : \mathcal{D} \cap \mathbb{Z}^d \to \mathbb{R}$, is $f_\Delta^-(x) := \min\{f(\tilde{x}) - f(x) : \tilde{x} \in \mathcal{N}_1(x)\}$ where $\mathcal{D} \subseteq \mathbb{R}^d$ is the domain of f.

Thus, if a point $x^* \in \mathcal{D} \cap \mathbb{Z}^d$ satisfies $f_\Delta^-(x) = 0$, then x^* is a local minimum of f. The definition for mildly-coercive uses the max-descent function.

Definition 3 A function $f\colon \mathcal{D} \cap \mathbb{Z}^d \to \mathbb{R}$ is said to be mildly-coercive if $\liminf_{k\to\infty} f_\Delta^-(x_k) < 0$ for any sequence $(x_k)_{k\geq 1} \subset \mathcal{D} \cap \mathbb{Z}^d$ such that $\|x_k\| \to \infty$.

Finally, we present several useful results that we invoke repeatedly to prove our results. First, we require Borel-Cantelli's first and second lemmas [5, p. 59], stated here as Lemma 1.

Lemma 1 *(Borel-Cantelli Lemmas) Let $(\mathcal{A}_n)_{n\geq 1}$ be a sequence of events defined on a probability space.*

1. *If $\sum_{n=1}^{\infty} P\{\mathcal{A}_n\} < \infty$, then $P\{\mathcal{A}_n \text{ i.o.}\} = 0$.*
2. *If events $\mathcal{A}_n$ are independent and $\sum_{n=1}^{\infty} P\{\mathcal{A}_n\} = \infty$, then $P\{\mathcal{A}_n \text{ i.o.}\} = 1$.*

We also require Lévy's Extension of Lemma 1 part 2 [51, p. 124], presented here as Lemma 2.

Lemma 2 *(Lévy's Extension of Borel-Cantelli) Let $(\mathcal{A}_n)_{n\geq 1}$ be a sequence of events defined on a filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_n)_{n\geq 0}, P)$, where $\mathcal{A}_n \in \mathcal{F}_n$ for each $n \geq 1$. If it holds that $\sum_{n=1}^{\infty} P\{\mathcal{A}_{n+1} \mid \mathcal{F}_n\} = \infty$, then $P\{\mathcal{A}_n \text{ i.o.}\} = 1$.*

Next, we require Bernstein's inequality [49, p. 33], stated here as Lemma 3, which allows us to bound the tail behavior of the estimated error.

Lemma 3 *(Bernstein Inequality) Let $X_1, X_2, \ldots, X_n$ be independent mean-zero sub-exponential random variables (see [49, p. 31] and Assumption 3). Then, for every $t \geq 0$,*

$$P\left\{\left|\sum_{i=1}^n X_i\right| \geq t\right\} \leq 2\exp\left\{-c_0 \min\left(\frac{t^2}{\sum_{i=1}^n \|X_i\|_{\psi_1}^2}, \frac{t}{\max_i \|X_i\|_{\psi_1}}\right)\right\},$$

where $c_0 > 0$ is an absolute constant, and the sub-exponential norm of a random variable X is $\|X\|_{\psi_1} \coloneqq \inf\left\{t > 0 \colon \mathbb{E}\left[\exp\{|X|/t\}\right] \leq 2\right\}$.

Finally, in Lemma 4, we present a simplified version of the Robbins-Siegmund Theorem [43], which provides conditions for an "almost-supermartingale."

Lemma 4 *(Simplified Robbins-Siegmund Theorem) Suppose $(Z_k)_{k\geq 0}$ and $(Y_k)_{k\geq 0}$ are positive-valued integrable stochastic processes, and let Z_k and Y_k be adapted to $\mathcal{F}_k$. Also, suppose $(Z_k)_{k\geq 0}$ is an almost-supermartingale, that is, for large enough k, $\mathbb{E}[Z_{k+1} \mid \mathcal{F}_k] \leq Z_k + Y_k$ w.p.1 and $\sum_{k=0}^{\infty} Y_k < \infty$ w.p.1. Then, $(Z_k)_{k\geq 0}$ converges w.p.1.*

2.2 Standing assumptions

The following standing assumptions pertain to the nature of the objective function f and the quality of the Monte Carlo estimator $\bar{F}$. First, we assume the objective function f is bounded below.

Assumption 1 The function $f\colon \mathcal{X} \to \mathbb{R}$ is such that $\inf_{x\in\mathcal{X}} f(x) > -\infty$.

Thus, the objective function f is finite-valued but does not necessarily attain its minimum. Next, we assume the objective function estimator is unbiased.

Assumption 2 Recall from (1) that $\varepsilon_i(\cdot) := F(\cdot, \xi_i) - f(\cdot), i = 1, 2, \ldots$. The sequence $(\xi_i)_{i \geq 1}$ is an i.i.d sequence with $\mathbb{E}[\varepsilon_i(x)] = 0$ for each $x \in \mathcal{X}$.

Then, if $\bar{\varepsilon}(\cdot, m) := m^{-1} \sum_{i=1}^{m} \varepsilon_i(\cdot)$, an unbiased estimator for f is $\bar{F}(\cdot, m) := f(\cdot) + \bar{\varepsilon}(\cdot, m)$. Recall that we observe $\bar{F}(\cdot, m)$ pointwise in $\mathcal{X}$; that is, we expend simulation effort m at the point x to obtain the value $\bar{F}(x, m)$.

Further, we assume the unbiased estimator is of sufficient quality.

Assumption 3 The error $\varepsilon_i(\cdot)$ is sub-exponential with sub-exponential norm $\kappa^* < \infty$; that is, $\kappa^* := \inf\left\{t > 0 \colon \mathbb{E}\left[\exp(\sup_{x \in \mathcal{X}} |\varepsilon_1(x)| / t)\right] \leq 2\right\} < \infty$.

Assumption 3 implies that the moment generating function of $\sup_{x \in \mathcal{X}} \varepsilon_1(x)$ exists and is finite in some open interval around $1/t = 0$. Further, Assumption 3 guarantees that for some universal constant $c > 0$ and for all $t \geq 0$,

$$P\left\{\sup_{x \in \mathcal{X}} \frac{1}{m} \sum_{i=1}^{m} |\varepsilon_i(x)| > t\right\} \leq P\left\{\frac{1}{m} \sum_{i=1}^{m} \sup_{x \in \mathcal{X}} |\varepsilon_i(x)| > t\right\} \leq 2 \exp\left(-\frac{mt}{c\kappa^*}\right), \quad (4)$$

where the second inequality in (4) follows from Lemma 3.

The next assumption pertains to how sampling is performed within ADALINE as the search continues across the feasible set $\mathcal{X}$ to generate the sequence of iterates $(X_k)_{k \geq 1}$. Assumption 4 formalizes the idea that the objective function estimates constructed at points visited during different iterations of ADALINE are independent; nevertheless, the assumption allows the algorithm to use what are called *common random numbers* (CRN) [25] at points visited during the same iteration; we discuss CRN further in §2.3. In Assumption 4, as is customary when rigorously analyzing random sequences generated by an algorithm, we assume the presence of a filtered space $(\Omega, \mathcal{F}, (\mathcal{F}_k)_{k \geq 0}, P)$, where $(\Omega, \mathcal{F}, P)$ is a probability triple as usual, and $(\mathcal{F}_k)_{k \geq 1}$ is a filtration, that is, an increasing family $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \cdots \subseteq \mathcal{F}$ of sub-σ-algebras of $\mathcal{F}$; see [51].

Assumption 4 Each iterate X_k is adapted to the sigma algebra $\mathcal{F}_k$ coming from a filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_k)_{k \geq 0}, P)$. Let $K = K(k) \in \mathcal{F}_k$ denote the (random) number of i.i.d. random fields sampled so far in obtaining iterate X_k. Then for all $k \geq 0$, the sequence $(\varepsilon_i)_{1 \leq i \leq K}$ is such that ε_i is $\mathcal{F}_k$-measurable for all $1 \leq i \leq K$; furthermore, for all $i > K$ and all $x \in \mathcal{X}$, we have $\mathbb{E}[\varepsilon_i(x) \mid \mathcal{F}_k] = 0$.

For some intuition surrounding Assumption 4, let us consider an example in which the last postulate would not hold. Suppose that in iteration $i > K$, loosely speaking, we reach into the algorithm's history and use a random field we have observed before. Then $\varepsilon_i \in \mathcal{F}_k$, which implies $\mathbb{E}[\varepsilon_i(x) \mid \mathcal{F}_k] = \varepsilon_i(x)$. If it further holds that $P\{\varepsilon_i(x) = 0\} = 0$ for all $x \in \mathcal{X}, i = 1, 2, \ldots$, then $\mathbb{E}[\varepsilon_i(x) \mid \mathcal{F}_k] = \varepsilon_i(x) \neq 0$ w.p.1.

2.3 Random fields and common random numbers

Within an iteration, ADALINE allows CRN, which is especially important during line search. To make the notion of CRN precise, suppose an iterative algorithm estimates

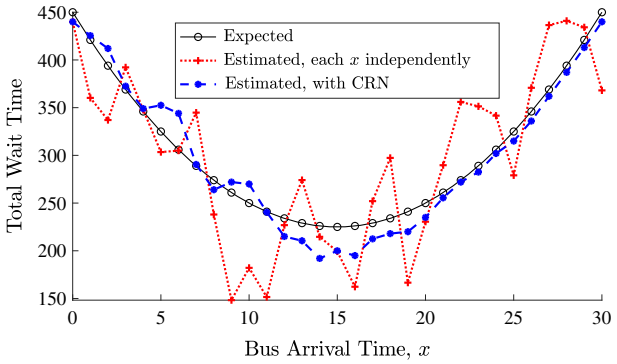

Fig. 1 The interpolated sample-path function for the one-bus scheduling problem [50] obtained with CRN (dashed line) has fewer local minima than the function that results from independently estimating the objective values at each $x \in \{0, 1, \ldots, 30\}$ (dotted line). Here, the arrival rate is $\lambda = 1$, and the number of observations is $m = 2$

the function f at the locations $x_1, x_2, \ldots, x_n \in \mathcal{X}$. Then, using CRN means using the *the same* random variables (or, on a computer, random variates) $\xi_i, i = 1, 2, \ldots, m$ to observe $\bar{F}(x_j, m)$, $j = 1, 2, \ldots, n$. Thus, *using CRN when estimating f at $x_1, x_2, \ldots, x_n$ means observing the random field $\bar{F}(\cdot, m) = m^{-1} \sum_{i=1}^{m} F(\cdot, \xi_i)$ at $x_1, x_2, \ldots, x_n$,* which is often called the *sample-path function*.

Depending on the application, CRN can be crucial to solving (Q) efficiently, since it retains any objective function structure that may be inherent. To see CRN in the context of a black-box SO problem, consider the discretized one-bus scheduling problem [50], which we re-visit in the numerical Sec. §5. In this problem, passengers arrive to a bus station according to a Poisson process with rate λ. A decision-maker would like to schedule a bus at a time $x \in [0, \tau] \cap \mathbb{Z}$ to minimize the expected total passenger wait time. Figure 1 shows the (interpolated) true function f, the sample-path function (obtained with CRN), and a corresponding estimated function obtained by independent simulation replications at each $x \in \{0, 1, \ldots, 30\}$. Notice that the sample-path function is "smoother" and has fewer $\mathcal{N}_1$-local minima than the corresponding estimated function obtained by independent sampling.

CRN implicitly underlies the construction (and success) of algorithms in a sample-average approximation (SAA) [20,45] framework. In contrast, a typical stochastic approximation or stochastic gradient descent [7,23,35] type algorithm that uses a mini-batch of size m to observe $\bar{F}(x_j, m)$, $j = 1, 2, \ldots, n$ does not use CRN, but often uses independent sampling, resulting in $\bar{F}(x_j, m)$, $j = 1, 2, \ldots, m$ being mutually independent.

3 The ADALINE algorithm for SO with integer variables

We now provide a high-level description of ADALINE. ADALINE repeatedly executes three key procedures to produce the stochastic sequence $(X_k)_{k \geq 0}$, where $X_0 \in \mathcal{X}$ is the initial guess and $X_{k+1} \in \mathcal{X}$, $k \geq 0$ is the iterate obtained upon the kth iteration's

Algorithm 1: ADALINE Algorithm for SO with Integer Variables

Input: Initial point $x_0 \in \mathcal{X}$; stochastic oracle $\bar{F}(\cdot, \cdot)$; minimum sample sizes $(\lambda_k)_{k \geq 0}$; significance level $\alpha \in (0, 1)$, number of obs. to obtain at once $\delta \in \mathbb{Z}^+$.

1 Initialize: $X_0 \leftarrow x_0$; $M_0 \leftarrow \lambda_0$; $N_0 \leftarrow 0$; search bound b_1; search bound $b_2 \leftarrow \sqrt{d} + 1$

2 **for** $k = 0, 1, 2, \ldots$ **do**

3 $\quad [\widetilde{X}_k, M_k] \leftarrow \mathrm{NE}(X_k, M_k^0, \lambda_k, \alpha, \delta)$ `/statistical optimality test`

4 $\quad \hat{d}_k \leftarrow (\widetilde{X}_k - X_k)/\|\widetilde{X}_k - X_k\|$ `/initial search direction`

5 $\quad N_k \leftarrow 1$

6 $\quad$ **repeat**

7 $\quad\quad \widetilde{X}_k \leftarrow \mathrm{LI}(\widetilde{X}_k, \hat{d}_k, M_k, b_1),\ N_k \leftarrow N_k + 1$ `/perform line search`

8 $\quad\quad$ **if** $N_k < b_2$ **then** $[\hat{d}_k, \widetilde{X}_k, \mathfrak{B}] \leftarrow \mathrm{DA}(\widetilde{X}_k, M_k, \lambda_k, \alpha, \delta)$ `/get new direction`

9 $\quad$ **until** $N_k \geq b_2$ *or* $\mathfrak{B}$ `/statistical optimality test needed`

10 $\quad X_{k+1} \leftarrow \widetilde{X}_k$; `/accept candidate iterate`

Note: On the initial iteration $k = 0$, we skip NE and begin with DA in step 8.

termination. Together with A for *adaptive*, the procedures form ADALINE's name: *direction acquisition* (DA), *line search* (LI), and *neighborhood enumeration* (NE). Next, we provide an overview of each.

First, each iteration k begins with Procedure NE, as listed in Algorithm 1. Procedure NE performs a statistical local optimality test at the current iterate X_k. If the statistical local optimality test fails at X_k, which corresponds to finding a statistically better neighbor $\widetilde{X}_k$, then Procedure NE terminates and returns $\widetilde{X}_k$ as the candidate next iterate. If the local optimality test does not fail, then Procedure NE does not terminate, and ADALINE's iterates are thus absorbed into a locally optimal point.

Assuming Procedure NE terminates with the better neighbor $\widetilde{X}_k$, ADALINE then executes multiple *line searches*. Each line search in the current context mimics the analogous line search procedure [36, Chapter 3] in the continuous context and proceeds as follows. Starting from $\widetilde{X}_k$, perform a one-dimensional search along a chosen direction by visiting points that are separated by increasing distances, and as long as the estimated objective values at the visited points continue to decrease. The estimated objective function values at points that do not lie on the integer lattice are substituted by objective function estimates at the corresponding nearest points on the lattice. The line searches are conducted by Procedure LI, initially along a vector pointing in the direction of the better neighbor, that is, $\widetilde{X}_k - X_k$, and subsequently in the stochastic descent direction identified by Procedure DA. Importantly, both Procedures LI and DA use the fixed sample size M_k, which facilitates the use of CRN in the objective function estimators. A formal listing of the line search procedure, including specifics on how points are chosen along the line, is included in Appendix A.

Procedure DA also returns a Boolean variable $\mathfrak{B}$ which is true if no better direction can be found at sample size M_k; that is, $\widetilde{X}_k$ is the estimated best point in its neighborhood at sample size M_k. In this case, iteration k concludes by accepting $\widetilde{X}_k$ as the next iterate X_{k+1}; iteration $k + 1$ commences with Procedure NE at X_{k+1}. If the Boolean variable $\mathfrak{B}$ is false and the limit on line searches has not been reached, then Procedure LI performs another line search along the direction identified by Procedure DA.

Given Procedure NE's centrality to ADALINE's efficiency and our asymptotic results, we provide an explanation and listing of Procedure NE first, in Sect. 3.1. In Sect. 3.2 and Sect. 3.3, we briefly discuss Procedures LI and DA, respectively. Detailed listings of Procedures LI and DA appear in the appendices.

3.1 Procedure NE: statistical local optimality test

Procedure NE (Algorithm 2) adaptively obtains function observations at points in the neighborhood $\mathcal{N}_1(X_k) \cap \mathcal{X}$ until it identifies a statistically better neighbor, $\widetilde{X}_k$. First, we describe the requirements for the better neighbor. Then, we describe the adaptive sampling choices that make Procedure NE efficient. In Sect. 4, we show that if X_k is not a local minimum, Procedure NE terminates w.p.1 under mild conditions.

3.1.1 Requirements for the better neighbor

Procedure NE imposes two requirements on the statistically better neighbor $\widetilde{X}_k$. First, the number of observations used to estimate the function value at $\widetilde{X}_k$ should be large enough, that is, at least as large as a lower bound specified by a constant λ_k. Second, whether $\widetilde{X}_k$ is statistically better than X_k is determined using a ratio that is reminiscent of a Student's t statistic.

To make these two requirements precise, we first define some notation. Let $M_k(x)$ denote the number of function observations obtained so far at each point $x \in \mathcal{N}_1(X_k) \cap \mathcal{X}$, where Procedure NE ensures $M_k(X_k) \geq M_k(x)$ for each $x \in \mathcal{N}_1(X_k) \cap \mathcal{X}$. Further, let the lower bound sample size sequence $(\lambda_k)_{k \geq 0} \subseteq \mathbb{Z}^+$ satisfy $\lambda_k^{-1}(\log k)^{1+\gamma} = O(1)$ for some constant $\gamma > 0$, which is a mild imposition on the rate of sample size increase that follows from the analysis in Sect. 4. Finally, let the constant $t_{\alpha,\nu}$ be the critical value of a Student's t distribution with ν degrees of freedom [18] where $\alpha \in (0, 0.5)$. Then, Procedure NE adaptively searches for a neighbor $\widetilde{X}_k \in \mathcal{N}_1(X_k) \cap \mathcal{X}$ satisfying

$$M_k := M_k(\widetilde{X}_k) \geq \lambda_k \tag{5a}$$

$$T_k(\widetilde{X}_k) \geq t_{\alpha,\nu_k} \tag{5b}$$

where $\nu_k = M_k(\widetilde{X}_k) - 1$ and

$$T_k(x) := \frac{\bar{F}(X_k, M_k(x)) - \bar{F}(x, M_k(x))}{\widehat{\text{s.e.}}\big(\bar{F}(X_k, M_k(x)) - \bar{F}(x, M_k(x))\big)} \quad \text{for } x \in \mathcal{N}_1(X_k). \tag{6}$$

Condition (5a) ensures that the sample size M_k exceeds the threshold λ_k and condition (5b) ensures that the difference in function estimates is significant in the sense of the estimated coefficient of variation, $T_k(\widetilde{X}_k)$, exceeding a fixed threshold. The estimated

Algorithm 2: $[\widetilde{X}_k, M_k] = \mathrm{NE}(X_k, \lambda_k, \alpha, \delta)$

Input: Center X_k; minimum sample size λ_k; significance level $\alpha \in (0, 1)$; number of observations $\delta \geq 1$ to obtain at once.
Output: Upon termination, better neighbor $\widetilde{X}_k \in \mathcal{N}_1'(X_k) \cap \mathcal{X}$, sample size M_k

Initialize: set of neighbors observed $\mathcal{A} \leftarrow \emptyset$; $\mu_* \leftarrow 0.001$
foreach $x \in \mathcal{N}_1'(X_k) \cap \mathcal{X}$ **do** $M_k(x) \leftarrow 0$, $T_k(x) \leftarrow 0$
repeat `/if X_k is an N_1-local minimizer, loop may not terminate`
  foreach $x \in \mathcal{N}_1'(X_k) \cap \mathcal{X}$ **do** update $\mu_k(x)$ using (8)
  calculate $\mathcal{A}_{\mathrm{low}} := \{x \in \mathcal{A}\colon M_k(x)/(\sum_{x \in \mathcal{A}} M_k(x)) \leq \mu_* |\mathcal{N}_1'(X_k) \cap \mathcal{X}|^{-1}\}$
  if $\mathcal{A}_{\mathrm{low}} \neq \emptyset$ **then** `/ensure all neighborhood points sampled i.o.`
    independently sample a point W^+ from $\mathcal{A}_{\mathrm{low}}$ with equal probability
  else
    use μ_k to independently sample a point W^+ from $\mathcal{N}_1'(X_k) \cap \mathcal{X}$
  $W^- \leftarrow X_k - (W^+ - X_k)$ `/get opposite neighbor point`
  obtain δ i.i.d. function observations at each of W^+ and W^- (using CRN)
  $M_k(W^+) \leftarrow M_k(W^+) + \delta$ and $M_k(W^-) \leftarrow M_k(W^-) + \delta$
  $\Delta^* \leftarrow \max\{(M_k(x) - M_k(X_k))^+ : x \in \mathcal{N}_1'(X_k) \cap \mathcal{X}\}$ `/calc. obs. at X_k`
  obtain Δ^* i.i.d. function observations at X_k (using CRN),
  $M_k(X_k) \leftarrow M_k(X_k) + \Delta^*$
  use (6) to update $T_k(W^+)$, $T_k(W^-)$.
until *there exists a point in* $\mathcal{N}_1'(X_k) \cap \mathcal{X}$ *that satisfies* (5a) *and* (5b)
$W^* \leftarrow \operatorname{argmax}\{T_k(X)\colon X \in \mathcal{N}_1'(X_k) \cap \mathcal{X}, X \text{ satisfies (5a) and (5b)}\}$
return $\widetilde{X}_k \leftarrow W^*$, $M_k \leftarrow M_k(W^*)$

standard error in equation (6) is defined as

$$\widehat{\mathrm{s.e.}}\big(\bar{F}(x_1, n) - \bar{F}(x_2, n)\big) := \sqrt{\frac{\hat{\sigma}_n^2(x_1, x_2)}{n}}$$

and $\hat{\sigma}_n^2(x_1, x_2) := n^{-1} \sum_{i=1}^n \big((F(x_1, \xi_i) - F(x_2, \xi_i)) - (\bar{F}(x_1, n) - \bar{F}(x_2, n)\big)^2$.

3.1.2 Adaptively sampling the neighborhood points

There are many ways to identify a better neighbor $\widetilde{X}_k$ satisfying the conditions in (5a) and (5b). Procedure NE accomplishes this task by adaptively sampling the neighborhood points according to a discrete probability measure μ_k supported on $\mathcal{N}_1'(X_k) \cap \mathcal{X}$ and updating the statistic $T_k(\cdot)$ until it identifies such a point. As the adaptive sampling progresses by obtaining δ function observations at a time, Procedure NE updates the discrete measure μ_k (Step 4) to guide the sampling effort toward points that are likely to be better than X_k. Judiciously updating the discrete measure μ_k is important for efficiency as it determines which neighbor is observed next. We now describe this step in further detail.

Procedure NE updates $\mu_k(x)$, $x \in \mathcal{N}_1'(X_k) \cap \mathcal{X}$ through a subjective probability calculation. Let the set $\mathcal{A} \subseteq \mathcal{N}_1'(X_k) \cap \mathcal{X}$ denote the set of neighbors of X_k whose estimated objective function values have already been observed with sample size greater than or equal to δ. Let $\mathcal{A}^c$ denote the set of unobserved neighbors of X_k; that

is, $\mathcal{A}^c := (\mathcal{N}_1'(X_k) \cap \mathcal{X}) \setminus \mathcal{A}$. Further, let

$$\mathcal{B} := \{x \in \mathcal{A} \colon \bar{F}(x, M_k(x)) < \bar{F}(X_k, M_k(x))\}$$

be the set of observed neighbors currently estimated as better than the incumbent iterate X_k, where each neighbor is compared to the incumbent iterate using its own same sample size and CRN. Then, defining $0/0 := 0$, we calculate the subjective probability of selecting a point from the observed set $\mathcal{A}$ as

$$\mu_k\{\mathcal{A}^c\} = \frac{p_0 \times |\mathcal{A}^c|}{|\mathcal{B}| + p_0 \times |\mathcal{A}^c|}; \quad \mu_k\{\mathcal{A}\} = 1 - \mu_k\{\mathcal{A}^c\}, \tag{7}$$

where $p_0 \in (0, 1)$ represents the subjective probability of an unobserved point having a better objective function value than the point X_k. Thus, p_0 represents the "exploration / exploitation" trade-off, and $p_0 \times |\mathcal{A}^c|$ is the subjective expected number of unobserved points in the neighborhood that are better than the incumbent X_k. (The default value $p_0 = 0.5$ is used within the solver and for all reported numerical experiments.)

Given (7), we calculate the individual subjective probabilities as

$$\mu_k(x) := \begin{cases} \mu_k\{\mathcal{A}^c\} \times |\mathcal{A}^c|^{-1} & x \in \mathcal{A}^c \\ \mu_k\{\mathcal{A}\} \times \frac{P\{T_{\nu(x)} > T_k(x)\}}{\sum_{x \in \mathcal{A}} P\{T_{\nu(x)} > T_k(x)\}} & x \in \mathcal{A}, \end{cases} \tag{8}$$

where $T_{\nu(x)}$ is a Student's t random variable with $\nu(x) = M_k(x)$ degrees of freedom and $T_k(\cdot)$ is the Student's t analogue in (6). The subjective discrete measure $\mu_k(\cdot)$ is thus the uniform measure conditional on the unobserved points, and is proportional to the Student's t analogue conditional on the already observed points.

Finally, notice that simulation replications are always obtained δ at a time, $\delta \in \mathbb{Z}^+$. The choice of δ should be made based on the time taken for each stochastic oracle call, the dimensionality d, and whether the oracle calls are performed in parallel [16]. While we can make robust choices for δ in practice, little is known theoretically about the relationship between the best value of δ, dimension d, and the time required to perform each stochastic oracle call.

3.2 Procedure LI: performing the line search

Procedure LI is a monotone-decreasing line search procedure that observes each point along the direction $\hat{d}_k$ with CRN at sample size M_k. Procedure LI terminates upon encountering a point whose estimated objective function value is inferior to the estimated objective function value at the previous point observed. In the first call to Procedure LI, the line search is performed along $\widetilde{X}_k - X_k$, that is, along the vector pointing in the direction of the best neighbor of X_k, as judged by Procedure NE. On subsequent calls, the direction is determined by Procedure DA. Procedure LI must also ensure that no more than b_1 points are visited during the line search, to prevent chase-offs when sample sizes are low. We make the requirements of Procedure LI precise in Sect. 4. A formal listing of Procedure LI appears in Sect. A.

3.3 Procedure DA: finding a direction

Recall that within an iteration of ADALINE, Procedure DA provides a direction to all but the first call to Procedure LI. To find a feasible direction that is also likely to be a descent direction, at a candidate iterate $\widetilde{X}_k$, Procedure DA first constructs an estimated descent cone. Then, Procedure DA constructs an estimated descent direction by weighting the directions in the estimated descent cone.

More specifically, first, Procedure DA attempts to locate $d+1$ feasible neighbors of $\widetilde{X}_k$ that are affinely independent. That is, these points are vertices of a randomly-oriented $(d+1)$-simplex formed by their convex hull; label these neighbors as $v_1, v_2, \ldots, v_s$, where $s \leq d+1$ and $s < d+1$ if Procedure DA cannot find $d+1$ such points. At each vertex, obtain M_k function observations and construct the objective function estimators $\bar{F}(v_j, M_k)$, $j = 1, 2, \ldots, s$. Then, the estimated descent cone $\mathcal{C}(\widetilde{X}_k)$ at $\widetilde{X}_k$ is the set of estimated descent directions, that is,

$$\mathcal{C}(\widetilde{X}_k) \coloneqq \{v_j - \widetilde{X}_k \colon \bar{F}(v_j, M_k) < \bar{F}(\widetilde{X}_k, M_k),\ j = 1, 2, \ldots, s\}.$$

If the estimated descent cone is nonempty, the estimated descent direction is a weighted combination of vectors in $\mathcal{C}(\widetilde{X}_k)$, where the weights are the Student's t analogue corresponding to each vector: $\hat{d}_k \coloneqq \sum_{\hat{d} \in \mathcal{C}(\widetilde{X}_k)} T_k(\hat{d} + X_k) \times \hat{d}$, where $T_k(\cdot)$ is the Student's t analogue in (6). Notice that when the point $\widetilde{X}_k$ lies on or close to the boundary of the feasible region, the descent cone $\mathcal{C}(\widetilde{X}_k)$ may be a small sub-space. This property is by design and has been observed to be important for good practical performance.

If Procedure DA attempts to construct the estimated descent cone $\mathcal{C}(\widetilde{X}_k)$ but cannot, then none of the neighbors of $\widetilde{X}_k$ have a better estimated objective function value than $\widetilde{X}_k$'s. Thus, Procedure DA terminates iteration k by setting the Boolean $\mathfrak{B}$ to true, so that ADALINE then accepts $\widetilde{X}_k$ as the next iterate X_{k+1}. A formal listing of Procedure DA appears in Sect. B.

4 Asymptotic analysis

We are now ready to analyze the behavior of ADALINE's iterates, $(X_k)_{k \geq 0}$. To do so, we require additional notation for the intermediate states visited between iterates X_k and X_{k+1}. The first intermediate state is the "better neighbor" of X_k found during Procedure NE, henceforth denoted $\widetilde{X}_{k,1}$. Other intermediate states may result as output from one or more calls to Procedure LI, henceforth denoted $\widetilde{X}_{k,2}, \ldots, \widetilde{X}_{k,N_k}$, where $N_k \leq b_2$ is the total number of intermediate states. The estimated objective function value at each intermediate state is observed with a sample size M_k. Recall from Assumption 4 that X_k is adapted to the sigma algebra $\mathcal{F}_k$ coming from a filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_k)_{k \geq 0}, P)$. For each $k \geq 0$ and all j, let $\widetilde{X}_{k,j}$, M_k, and N_k be $\mathcal{F}_k$-measurable. Figure 2 illustrates the states and intermediate states.

In case Procedure NE does not terminate in some iteration k, such that the residence time in X_k is infinite, then for all $k' \geq k+1$, let $X_{k'} \coloneqq X_k$. Further, let $N_{k'} = 1$ and

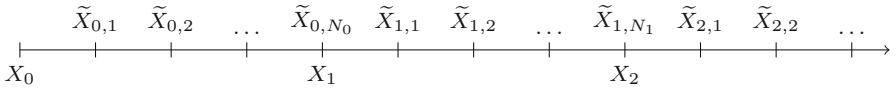

Fig. 2 The states $X_0, X_1, X_2, \ldots$ are the locations where ADALINE performs the statistical optimality test via Procedure NE. Between states X_k and X_{k+1}, $k \geq 0$, ADALINE visits $N_k \geq 1$ intermediary points $\widetilde{X}_{k,j}$, $j = 1, 2, \ldots N_k$. The point $\widetilde{X}_{k,1}$ is the "better neighbor" of X_k from Procedure NE. The remaining points result from calls to Procedure LI

$\widetilde{X}_{k',1} \coloneqq X_k$ for all $k' \geq k+1$. In this way, we ensure the better neighbors from NE, denoted by $\widetilde{X}_{k',1}$, are defined as equal to X_k.

The following Lemma 5 asserts that in Procedure NE, ADALINE "escapes" an iterate X_k that is not a local minimum w.p.1. Recall that the set of all $\mathcal{N}_1$-local minimizers is $\mathcal{L}(\mathcal{N}_1) \coloneqq \{x^* \in \mathcal{X}\colon f(x^*) \leq f(x) \text{ for all } x \in \mathcal{N}_1(x^*) \cap \mathcal{X}\}$.

Lemma 5 *(Escape Lemma) If $x \notin \mathcal{L}(\mathcal{N}_1)$, then for each $k \geq 0$, we have $P\{X_{k'} = x, \widetilde{X}_{k',1} = x \text{ for all } k' \geq k+1 \mid X_k = x, \mathcal{F}_k\} = 0$ w.p.1*

Proof (Sketch) For a contradiction, let $x \notin \mathcal{L}(\mathcal{N}_1)$, and suppose that there exists $k \geq 0$ such that $X_{k'} = x$, $\widetilde{X}_{k',1} = x$ for all $k' \geq k+1$, given $X_k = x, \mathcal{F}_k$. Since $x \notin \mathcal{L}(\mathcal{N}_1)$, there exists a better neighborhood point $\tilde{x} \in \mathcal{N}_1(x)$ such that $f(\tilde{x}) < f(x)$. Let $\tilde{x}_*$ be the "worst" of these, $\tilde{x}_* = \text{argmax}\{f(\tilde{x})\colon \tilde{x} \in \mathcal{N}_1(x),\ f(\tilde{x}) < f(x)\}$. Let $\kappa_* \coloneqq f(x) - f(\tilde{x}_*) > 0$. Since Procedure NE does not terminate, there must never be a neighborhood point that satisfies both (5a) and (5b). However, notice there exists a random but finite total number of function observations beyond which $\tilde{x}_*$ satisfies all conditions w.p.1. (From Step 6 in Algorithm 2, the parameter $\mu_* > 0$ ensures that each neighborhood point is sampled i.o. when Procedure NE is non-terminating.) □

Along with our assumptions in Sect. 2.2, the following three properties of ADALINE, noted in Sect. 3 and written precisely here, are important for its convergence and convergence rate characteristics.

Property 1 (Sufficient Decrease in Procedure NE) For large enough k, given that X_k is not a local minimum, there is positive probability that Procedure NE returns a "best" neighborhood point of X_k as the better neighbor $\widetilde{X}_{k,1}$ w.p.1. That is, there exists $\kappa_1 > 0$ such that for large enough k, $P\left\{\widetilde{X}_{k,1} \in \text{argmin}\{f(x)\colon x \in \mathcal{N}_1(X_k)\} \,\middle|\, X_k \notin \mathcal{L}(\mathcal{N}_1),\ \mathcal{F}_k\right\} \geq \kappa_1$ w.p.1

Property 2 (Monotone Line Search) For each iteration $k \geq 0$ and Procedure LI call that results in intermediate point $\widetilde{X}_{k,j}$ with sample size M_k, let the points visited by Procedure LI in iteration k and line search j be denoted $W_{k,j,0} \coloneqq \widetilde{X}_{k,j-1}, W_{k,j,1}, W_{k,j,2}, \ldots, W_{k,j,L_{k,j}} \coloneqq \widetilde{X}_{k,j}$, where $L_{k,j} \leq b_1$ is the total number of better points visited in Procedure LI. The line search ensures

$$\bar{F}(W_{k,j,0}, M_k) \geq \bar{F}(W_{k,j,1}, M_k) \geq \ldots \geq \bar{F}(W_{k,j,L_{k,j}}, M_k) \text{ w.p.1.}$$

Property 3 (Finite Searches) The iterates satisfy $\sup_{k \geq 0} \|X_k - X_{k+1}\| < \infty$.

Property 1, connoting sufficient decrease following the statistical optimality test, holds by the logic of Procedure NE and the Escape Lemma (Lemma 5). From a ranking and selection standpoint, this property corresponds to using a procedure whose probability of false selection is strictly less than one — a very mild condition. Property 2 holds because the line search logic in Procedure LI ensures that the function estimates are non-increasing until the penultimate visited point, which is accepted as a candidate iterate. Finally, recall that due to the limits on the line search length b_1 and number of line searches b_2, no more than $b_1 \times b_2$ points are visited by ADALINE on its way from X_k to X_{k+1} for all $k \geq 0$. Thus, $N_k \leq b_1 \times b_2$ for all $k \geq 0$, which ensures Property 3 holds.

Using these properties, we now show that the function estimate sequence at ADALINE's iterates converge w.p.1 to the corresponding sequence of true function values, as long as the lower bound sample size sequence λ_k increases faster than logarithmically.

Lemma 6 *Suppose the lower bound sample size sequence* $(\lambda_k)_{k\geq 0}$ *increases faster than logarithmically. That is,* $(\lambda_k)_{k\geq 0}$ *is such that, for some* $\gamma > 0$*,*

$$\lambda_k^{-1}(\log k)^{1+\gamma} = O(1). \tag{9}$$

Recall that $M_k \geq \lambda_k$ *denotes the exit sample size from Procedure NE condition* (5a)*. Then the following hold:*

1. *As* $k \to \infty$, $\bar{F}(X_k, M_k) - f(X_k) \to 0$ *w.p.1;*
2. *As* $k \to \infty$, $\bar{F}(\widetilde{X}_{k,j}, M_k) - f(\widetilde{X}_{k,j}) \to 0$ *w.p.1;*

Proof *(Part 1)* Let $X_k \in \mathcal{X}$ be the current iterate. Then for every $t \geq 0$,

$$\begin{aligned} P\left\{\bar{F}(X_k, M_k) - f(X_k) > t\right\} &\leq P\left\{\sup_{m\geq\lambda_k} \bar{F}(X_k, m) - f(X_k) > t\right\} \\ &\leq \sum_{m=\lambda_k}^{\infty} \mathbb{P}\left\{\bar{F}(X_k, m) - f(X_k) > t\right\} \leq \sum_{m=\lambda_k}^{\infty} P\left\{\sup_{x\in\mathcal{X}} \bar{\varepsilon}_i(x, m) > t\right\} \\ &\leq \sum_{m=\lambda_k}^{\infty} 2\exp\left(-\frac{mt}{c\kappa^*}\right) = \frac{2\exp\left(-\lambda_k t/(c\kappa^*)\right)}{1-\exp\left(-t/(c\kappa^*)\right)} \end{aligned} \tag{10}$$

where the inequality in (10) follows from (4) and the equality in the same line follows because we have the sum of a geometric series.

Let $\kappa_a := t/(c\,\kappa^*)$ and $\kappa_b := 2(1-\exp(-\kappa_a))^{-1}$. Since $(\lambda_k)_{k\geq 0}$ satisfies the sample size increase condition in (9), we have

$$\sum_{k=0}^{\infty} P\left\{\bar{F}(X_k, M_k) - f(X_k) > t\right\} \leq \kappa_b \sum_{k=0}^{\infty} \exp\left(-\lambda_k \kappa_a\right) < \infty. \tag{11}$$

Using the first Borel-Cantelli lemma (Lemma 1) on (11) concludes the proof.

(Part 2) We omit the proof, which follows from similar arguments. □

Using the results proved thus far, we are now ready to characterize the convergence and convergence rate of ADALINE. In the following Theorem 1 part (1), we prove that ADALINE's iterates form an almost-supermartingale process in the sense of Lemma 4. This per se does not guarantee that $(X_k)_{k\geq 0}$ converges to a local minimum; we prove this fact in Theorem 1 part (2).

Theorem 1 *Suppose the lower bound sample size sequence $(\lambda_k)_{k\geq 0}$ is such that, for some $\gamma > 0$, $\lambda_k^{-1} k^{1+\gamma} = O(1)$. Then the following assertions hold.*

1. *The function value sequence $(f(X_k))_{k\geq 0}$ is an almost-supermartingale. That is, there exists a positive-valued integrable sequence $(Y_k)_{k\geq 0}$ such that for large enough k, $\mathbb{E}[f(X_{k+1}) \mid \mathcal{F}_k] \leq f(X_k) + Y_k$ w.p.1 and $\sum_{k=1}^{\infty} Y_k < \infty$ w.p.1*
2. *The max-descent function sequence $(f_{\Delta}^{-}(X_k))_{k\geq 0}$ (see Definition 2) satisfies $\lim_{k\to\infty} f_{\Delta}^{-}(X_k) = 0$ w.p.1*

Proof *(Part 1)* Under Assumption 1, f is bounded below. Let $v_* := \inf_{x\in\mathcal{X}} f(x)$, $v_* > -\infty$, denote the infimum of f, so that $G_k := f(X_k) - v_*$ is the non-negative optimality gap of X_k. Recall that $X_k, \widetilde{X}_{k,1}, \widetilde{X}_{k,2}, \ldots, \widetilde{X}_{k,N_k} = X_{k+1}$ are the points visited by ADALINE's iteration k, where $N_k \leq b_2$ is finite. Let $\widetilde{X}_{k,0} := X_k$ and write

$$f(X_{k+1}) - f(X_k) = f(\widetilde{X}_{k,1}) - f(X_k) + \textstyle\sum_{j=2}^{N_k} \big(f(\widetilde{X}_{k,j}) - f(\widetilde{X}_{k,j-1})\big). \quad (12)$$

Recall $\bar{\varepsilon}(x,m) := m^{-1}\sum_{i=1}^{m} \varepsilon_i(x) = \bar{F}(x,m) - f(x)$ for each $x \in \mathcal{X}$. After appropriate algebra, (12), condition (5), and Property 2 imply that w.p.1,

$$\begin{aligned}
&G_{k+1} - G_k \\
&= \bar{\varepsilon}(X_k, M_k) - \bar{\varepsilon}(\widetilde{X}_{k,1}, M_k) + \big(\textstyle\sum_{j=2}^{N_k} \bar{\varepsilon}(\widetilde{X}_{k,j-1}, M_k) - \bar{\varepsilon}(\widetilde{X}_{k,j}, M_k)\big) \\
&\quad + \bar{F}(\widetilde{X}_{k,1}, M_k) - \bar{F}(X_k, M_k) + \big(\textstyle\sum_{j=2}^{N_k} \bar{F}(\widetilde{X}_{k,j}, M_k) - \bar{F}(\widetilde{X}_{k,j-1}, M_k)\big) \\
&\leq \bar{\varepsilon}(X_k, M_k) - \bar{\varepsilon}(\widetilde{X}_{k,1}, M_k) + \big(\textstyle\sum_{j=2}^{N_k} \bar{\varepsilon}(\widetilde{X}_{k,j-1}, M_k) - \bar{\varepsilon}(\widetilde{X}_{k,j}, M_k)\big) \\
&= \big(\textstyle\sum_{j=0}^{N_k-1} \bar{\varepsilon}(\widetilde{X}_{k,j}, M_k)\big) - \bar{\varepsilon}(\widetilde{X}_{k,N_k}, M_k) - \textstyle\sum_{j=2}^{N_k} \bar{\varepsilon}(\widetilde{X}_{k,j-1}, M_k) \\
&= \bar{\varepsilon}(\widetilde{X}_{k,0}, M_k) - \bar{\varepsilon}(\widetilde{X}_{k,N_k}, M_k). \qquad (13)
\end{aligned}$$

From (4), for all $j \in \{0, 1, 2, \ldots, N_k\}$, all k, and all $t \geq 0$,

$$P\big\{\,\big|\bar{\varepsilon}(\widetilde{X}_{k,j}, M_k)\big| > t\big\} \leq 2\exp\big(-\lambda_k t/(c\kappa^*)\big). \quad (14)$$

Integrating (14) with respect to t, for all $j \in \{0, 1, 2, \ldots, N_k\}$ and all k,

$$\mathbb{E}\big[\big|\bar{\varepsilon}(\widetilde{X}_{k,j}, M_k)\big|\big] \leq 2c\kappa^*\lambda_k^{-1}; \quad (15)$$

thus, $|\bar{\varepsilon}(\widetilde{X}_{k,j}, M_k)|$ is integrable. Under the required sample size increase rate, for each $j \in \{0, 1, 2, \ldots, N_k\}$,

$$\sum_{k=0}^{\infty} \mathbb{E}\big[\big|\bar{\varepsilon}(\widetilde{X}_{k,j}, M_k)\big|\big] = \mathbb{E}\big[\sum_{k=0}^{\infty} \big|\bar{\varepsilon}(\widetilde{X}_{k,j}, M_k)\big|\big] \leq \sum_{k=0}^{\infty} 2c\kappa^*\lambda_k^{-1} < \infty, \quad (16)$$

which implies $\sum_{k=0}^{\infty} \big|\bar{\varepsilon}(\widetilde{X}_{k,j}, M_k)\big| < \infty$. Notice that the above arguments hold if we replace $\bar{\varepsilon}(\widetilde{X}_{k,j}, M_{k,j})$ with $\bar{\varepsilon}(\widetilde{X}_{k,j-1}, M_{k,j})$.

From equation (13), using Assumption 4 and recalling that for all j the random variables $\widetilde{X}_{k,j}$, M_k, and N_k are $\mathcal{F}_k$-measurable, we have

$$\begin{aligned}\mathbb{E}\left[f(X_{k+1}) \,|\, \mathcal{F}_k\right] - f(X_k) &= \mathbb{E}\left[G_{k+1} - G_k \,|\, \mathcal{F}_k\right] \\ &\leq |\bar{\varepsilon}(\widetilde{X}_{k,0}, M_k)| + |\bar{\varepsilon}(\widetilde{X}_{k,N_k}, M_k)| \quad \text{w.p.1.} \end{aligned} \tag{17}$$

Since the right-hand side of (17) is finite w.p.1, along with Lemma 6 and (16), imply $(f(X_k) - v_*)_{k\geq 0}$ is a positive-valued almost-supermartingale.

(Part 2) Since $(f(X_k) - v_*)_{k\geq 0}$ is a positive-valued almost-supermartingale, Lemma 4 implies $(f(X_k) - v_*)_{k\geq 0}$ converges w.p.1, hence $(f(X_k))_{k\geq 0}$ converges w.p.1 Now, for a contradiction, suppose that with positive probability the max-descent function sequence $(f_\Delta^-(X_k))_{k\geq 0}$ does not tend to zero. Then, with positive probability, there exists $\epsilon > 0$ and a sub-sequence $(k_\ell)_{\ell\geq 1}$ such that $f_\Delta^-(X_{k_\ell}) < -\epsilon$; that is, on the subsequence, X_{k_ℓ} has a neighbor that is at least ϵ-better, so that $\min\{f(\tilde{x}) - f(X_{k_\ell}) \colon \tilde{x} \in \mathcal{N}_1(X_{k_\ell})\} < -\epsilon$ for all $\ell \geq 1$. Then,

$$\begin{aligned}\textstyle\sum_{i=0}^{\infty} P\{f(X_{i+1}) - f(X_i) \leq -\epsilon \,|\, \mathcal{F}_i\} \\ \geq \textstyle\sum_{\ell=1}^{\infty} P\{f(X_{k_\ell+1}) - f(X_{k_\ell}) \leq -\epsilon \,|\, \mathcal{F}_{k_\ell}\} \\ \geq \textstyle\sum_{\ell=1}^{\infty} P\left\{f(X_{k_\ell+1}) \leq \min\{f(x) \colon x \in \mathcal{N}_1(X_{k_\ell})\} \,|\, \mathcal{F}_{k_\ell}\right\} \\ \geq \textstyle\sum_{\ell=1}^{\infty} \kappa_1 = \infty, \end{aligned} \tag{18}$$

where (18) follows from Properties 1 and 2 . The filtered version of the Borel-Cantelli lemma (Lemma 2) implies that w.p.1, $P\{f(X_{i+1}) - f(X_i) \leq -\epsilon \text{ i.o.}\} = 1$, which contradicts the fact that $f(X_k)$ is bounded from below. □

The second part of Theorem 1 is analogous to convergence proofs in the smooth deterministic context [34] which demonstrate that $\|\nabla f(x_k)\| \to 0$. Importantly, Theorem 1 says nothing about whether ADALINE's iterates attain a local minimum; it only states that ADALINE's iterates approach a region with zero max-descent in the limit w.p.1. This result is the best one can prove since the assumptions imposed on f do not preclude situations where the max-descent function tends to zero along sequences $(X_k)_{k\geq 0}$ that diverge.

We now state a stronger theorem that asserts absorption into the set of local minima under the restriction that the function f is *mildly-coercive* (Definition 3). Moreover, we show that such absorption happens exponentially fast in iteration number. Theorem 2 does not, however, imply absorption into a singleton within $\mathcal{L}(\mathcal{N}_1)$.

Theorem 2 *(If f is mildly-coercive, ADALINE's iterates are absorbed into the set of local minima $\mathcal{L}(\mathcal{N}_1)$ in finite time.) Let the postulates of Theorem 1 hold, and let f be mildly-coercive. Then the sequence $(X_k)_{k\geq 0}$ of ADALINE's iterates is such that for large enough k, $X_k \in \mathcal{L}(\mathcal{N}_1)$ w.p.1*

Proof Under the postulates of Theorem 1, $(f_\Delta^-(X_k))_{k\geq 0} \to 0$ w.p.1 Furthermore, f is mildly-coercive. Then, there exists a finite set $\mathcal{W}$, independent of $\omega \in \Omega$, such that $X_k \in \mathcal{W}$ for large enough k w.p.1 Now, for a contradiction, suppose that $(X_k)_{k\geq 0}$ does not become absorbed in the set of local minima, that is, with positive probability, there

exists a subsequence $(X_{k_\ell})_{\ell \geq 1}$ such that $X_{k_\ell} \notin \mathcal{L}(\mathcal{N}_1)$. Since $\mathcal{W}$ is finite, there exists $\epsilon' > 0$ such that $\liminf_{\ell \to \infty} f_\Delta^-(X_{k_\ell}) < -\epsilon'$. Then, the theorem holds by the same proof as Theorem 1 part (2). □

Next, we show that the total number of stochastic oracle calls made before absorption into the set of local minima is finite w.p.1. Furthermore, the probability of non-absorption into the set of local minima decays exponentially fast in the number of stochastic oracle calls.

Theorem 3 *(The probability of ADALINE's iterates remaining unabsorbed into the set of local minimizers decays exponentially fast in iteration number.) Suppose the lower bound sample size sequence $(\lambda_k)_{k \geq 1}$ is such that $\lambda_k^{-1} k^{1+\gamma} = O(1)$ for some $\gamma > 0$, so that for large enough k, there exists $c_2 \in (0, \infty)$ such that $\lambda_k > k^{1+\gamma} c_2^{-1}$. Also, let f be mildy-coercive (see Definition 3). Then for characterizable constant $\kappa_2 > 0$ and large enough iteration number k_0,*

$$P\left\{\bigcup_{k=k_0}^{\infty} X_k \notin \mathcal{L}(\mathcal{N}_1)\right\} \leq \frac{16bcc_2\kappa^*}{\kappa_2} \exp\left(-\frac{(k_0-1)^{1+\gamma}\kappa_2}{2bcc_2\kappa^*}\right).$$

Proof Let the (good) event the process $(X_k)_{k \geq 0}$ lives in the local set forever after iteration k_0 be $\mathcal{A}_{k_0} := \cap_{k=k_0}^{\infty}\big(X_k \in \mathcal{L}(\mathcal{N}_1)\big)$. Further, let the (bad) events that there are an infinite number of "jumps up" and "jumps down" after iteration k_0, respectively, be $\mathcal{B}_{k_0}^{\text{up}} := \cap_{k=k_0}^{\infty} \cup_{j=k}^{\infty} \big(f(X_j) < f(X_{j+1})\big)$ and $\mathcal{B}_{k_0}^{\text{down}} := \cap_{k=k_0}^{\infty} \cup_{j=k}^{\infty} \big(f(X_j) > f(X_{j+1})\big)$. Finally, let $\mathcal{B}_{k_0}^{\text{same}}$ denote the (bad) event that the process remains in the some non-local point forever after iteration k_0, $\mathcal{B}_{k_0}^{\text{same}} := \cup_{x \notin \mathcal{L}(\mathcal{N}_1)} \cap_{k=k_0}^{\infty} (f(X_k) = f(x))$. Notice that

$$P\{\mathcal{A}_{k_0}^c\} \leq P\{\mathcal{B}_{k_0}^{\text{up}}\} + P\{\mathcal{B}_{k_0}^{\text{down}}\} + P\{\mathcal{B}_{k_0}^{\text{same}}\}. \tag{19}$$

We now quantify the probabilities of each event on the right side of (19), beginning with $\mathcal{B}_{k_0}^{\text{up}}$. From the proof of Theorem 2, for large enough k, the sequence $(X_k)_{k \geq 1}$ remains within a fixed and bounded set $\mathcal{W}$ w.p.1. Thus, there exists $\kappa_2 > 0$ such that $|f(x_1) - f(x_2)| > \kappa_2$ for all $x_1, x_2 \in \mathcal{W}$ such that $f(x_1) \neq f(x_2)$. Then, using (13), for large enough $k_0 \geq 2$,

$$\begin{aligned}
P\{\mathcal{B}_{k_0}^{\text{up}}\} &\leq \textstyle\sum_{k=k_0}^{\infty} P\{f(X_{k+1}) - f(X_k) > 0\} \\
&\leq \sum_{k=k_0}^{\infty} P\Big\{|\bar{\varepsilon}(\widetilde{X}_{k,0}, M_k)| > \frac{\kappa_2}{2b}\Big\} + P\Big\{|\bar{\varepsilon}(\widetilde{X}_{k,N_k}, M_k)| > \frac{\kappa_2}{2b}\Big\} \\
&\leq \sum_{k=k_0}^{\infty} 4\exp\left(-\frac{\lambda_k \kappa_2}{2bc\kappa^*}\right) \leq \sum_{k=k_0}^{\infty} 4\exp\big(-\frac{k^{1+\gamma}\kappa_2}{2bcc_2\kappa^*}\big) \qquad (20) \\
&\leq \int_{k_0-1}^{\infty} 4\exp\big(-\frac{x^{1+\gamma}\kappa_2}{2bcc_2\kappa^*}\big)dx \leq \frac{8bcc_2\kappa^*}{\kappa_2}\exp\big(-\frac{(k_0-1)^{1+\gamma}\kappa_2}{2bcc_2\kappa^*}\big), \qquad (21)
\end{aligned}$$

where the first inequality in (20) is from (4). Also, for large enough k_0,

$$P\{\mathcal{B}_{k_0}^{\text{down}}\} = P\{\mathcal{B}_{k_0}^{\text{down}} \cap \mathcal{B}_{k_0}^{\text{up}}\} + P\{\mathcal{B}_{k_0}^{\text{down}} \cap (\mathcal{B}_{k_0}^{\text{up}})^c\}$$
$$\leq P\{\mathcal{B}_{k_0}^{\text{down}} \cap \mathcal{B}_{k_0}^{\text{up}}\} + 0 \leq \frac{8bcc_2\kappa^*}{\kappa_2} \exp\Big(-\frac{(k_0-1)^{1+\gamma}\kappa_2}{2bcc_2\kappa^*}\Big), \tag{22}$$

where the 0 before the second inequality in (22) follows because, since the function is bounded below, it is impossible to have an infinite number of jumps down while having only a finite number of jumps up. The second inequality in (22) follows from (21). Considering the event $\mathcal{B}_{k_0}^{\text{same}}$, by Property 1, $P\{\mathcal{B}_{k_0}^{\text{same}}\} = 0$. Substitute this result, along with the results in (21) and (22), into the expression in (19) to yield the result. □

Theorem 3 is interesting but does not provide a full sense of the efficiency of the algorithm because the convergence rate in this theorem has been expressed in terms of the number of iterations as opposed to the number of oracle calls. This inadequacy of iteration complexity as a measure of efficiency is evident in that, thus far, we have imposed no upper bound on the rate of increase of the sample sizes. In the result that follows, we thus attempt to capture the efficiency of the proposed algorithm through a work complexity result that expresses the convergence rate in terms of the total number of oracle calls.

Theorem 4 *Let the postulates of Theorem 3 hold, and let the expected number of stochastic oracle calls expended by Procedure NE during iteration k be denoted* $V_k := \sum_{x\in\mathcal{N}_1(X_k)\cap\mathcal{X}} \mathbb{E}[M_k(x) \mid \mathcal{F}_k]$. *Then, there exists a constant* $\Lambda < \infty$ *such that for large enough k,*

$$V_k \, \mathbb{I}\{X_k \notin \mathcal{L}(\mathcal{N}_1)\} \leq (\lambda_k + \Lambda)\, d \;\; w.p.1. \tag{23}$$

Further, let $\lambda_k \leq \gamma_0(\log k)^{1+\gamma_1}$ *for some* $\gamma_0, \gamma_1 > 0$. *Then the total expected number of oracle calls outside the set of local minima and after iteration* k_0, $W_{k_0} := \sum_{k=k_0}^{\infty} \mathbb{E}\,[V_k \, \mathbb{I}\{X_k \notin \mathcal{L}(\mathcal{N}_1)\}]$, *satisfies, for some* $\tilde{\Lambda} < \infty$,

$$W_{k_0} \leq \lambda_{k_0}\, \tilde{\Lambda}\, d \quad \text{as } k_0 \to \infty. \tag{24}$$

Proof Since (23) is trivially true if $X_k \in \mathcal{L}(\mathcal{N}_1)$, let's suppose $X_k \notin \mathcal{L}(\mathcal{N}_1)$. We know from arguments in the proof of Theorem 2 that there exists a finite set $\mathcal{W}$, independent of $\omega \in \Omega$, such that $X_k \in \mathcal{W}$ for large enough k w.p.1 Consequently, there exists $\delta > 0$ such that the "best" neighbor of X_k is at least δ better, that is, $f(X_k) - \min\{f(x) : x \in \mathcal{N}_1(X_k)\} \geq \delta > 0$. For $x \in \operatorname{argmin}\{f(x) : x \in \mathcal{N}_1(X_k)\}$, notice that

$$\begin{aligned}
\mathbb{E}[M_k(x) \mid \mathcal{F}_k] &= \textstyle\sum_{m=1}^{\infty} P\{M_k(x) > m \mid \mathcal{F}_k\} \\
&= \lambda_k + \textstyle\sum_{m=\lambda_k}^{\infty} P\{\bar{F}(X_k, m) \leq \bar{F}(x, m) \mid \mathcal{F}_k\} \\
&\leq \lambda_k + \textstyle\sum_{m=\lambda_k}^{\infty} P\{\bar{F}(X_k, m) - f(X_k) \leq \bar{F}(x, m) - f(x) - \delta \mid \mathcal{F}_k\} \\
&\leq \lambda_k + \textstyle\sum_{m=\lambda_k}^{\infty} P\{m^{-1} \textstyle\sum_{i=1}^{m} \sup_{x\in\mathcal{X}} |\varepsilon_i(x)| \geq \tfrac{\delta}{2} \mid \mathcal{F}_k\}
\end{aligned}$$

$$\leq \lambda_k + \sum_{m=\lambda_k}^{\infty} 2\exp\left(-\frac{m\delta}{2c\kappa^*}\right) \leq \lambda_k + \Lambda, \tag{25}$$

where the penultimate inequality in (25) follows from Assumption 3 and $\infty > \Lambda := \sum_{m=0}^{\infty} 2\exp\left(-\frac{m\delta}{2c\kappa^*}\right) > \sum_{m=\lambda_k}^{\infty} 2\exp\left(-\frac{m\delta}{2c\kappa^*}\right)$. Conclude from (25) and $|\mathcal{N}_1(X_k) \cap \mathcal{X}| \leq d$ that (23) holds.

To see that (24) holds, combine (25) with Theorem 2. □

Theorem 4 makes three important assertions pertaining to the oracle efficiency of the proposed algorithm. First, the bound in (23) asserts that the total expected oracle calls spent at a sub-optimal point is almost surely bounded by the lower bound sample size λ_k for large k. Second, the asymptotic inequality in (24) asserts that the total work done (past a large iteration k_0) at sub-optimal points is bounded by the lower bound sample size at k_0. These two inequalities point to choosing a λ_k according to the minimum lower bound stipulated through (9). Third, both inequalities (23) and (24) indicate a linear dependence on dimension d.

5 Numerical illustration

In this section, we illustrate the performance of ADALINE on a variety of problems within two problem suites: bus scheduling problems and ill-conditioned discrete quadratic problems.

5.1 Bus scheduling problem suite

Suppose passengers arrive to a bus depot according to a homogeneous Poisson process with rate λ. We wish to schedule d buses in a fixed time interval $[0, \tau]$ to minimize the total expected wait time of all passengers arriving in the interval. We assume the following: (a) each bus has infinite capacity; (b) pre-scheduled buses depart at time 0 and at time τ; (c) $\lambda = 10$, and (d) $\tau = 100$. The decision variable is the departure times of the d buses, $(x_1, \cdots, x_d)$, where each departure must occur in the interval $[0, \tau]$. We consider any of the $d!$ permutations of $(x_1, \cdots, x_d)$ to be equivalent. For simplicity, henceforth, label the buses so that $x_0 := 0 \leq x_1 \leq x_2 \leq \ldots \leq x_d \leq x_{d+1} := \tau$.

While the simulation oracle only has access to the random passenger arrival times on each simulation replication, for comparing algorithmic performance, we calculate the total expected wait time as $g(x) = (\lambda/2)\sum_{i=1}^{d+1}(x_i - x_{i-1})^2$. When $\text{mod}(\tau, d+1) = 0$, the optimal bus schedule x^* has buses departing every $\tau/(d+1)$ time units, with a corresponding minimum expected wait time of $g(x^*) = \lambda\tau^2/(2(d+1))$. For example, in the 9-bus scheduling problem, the solution is one of the 9! permutations of $x^* = (10, 20, 30, 40, 50, 60, 70, 80, 90)$, and the total expected wait time is 5000 time units.

We compare the performances of ADALINE and R-SPLINE on the 9, 20, 50, and 100 bus scheduling problems in Fig. 3. Our comparison is based on 1000 independent replications of each algorithm, starting from the same initial point $x_0 = (1, \ldots, 1) \in \mathbb{R}^d$. The algorithm parameters in both ADALINE and R-SPLINE are fixed ahead of time, without tuning. For example, in R-SPLINE, the sample size increase rate was

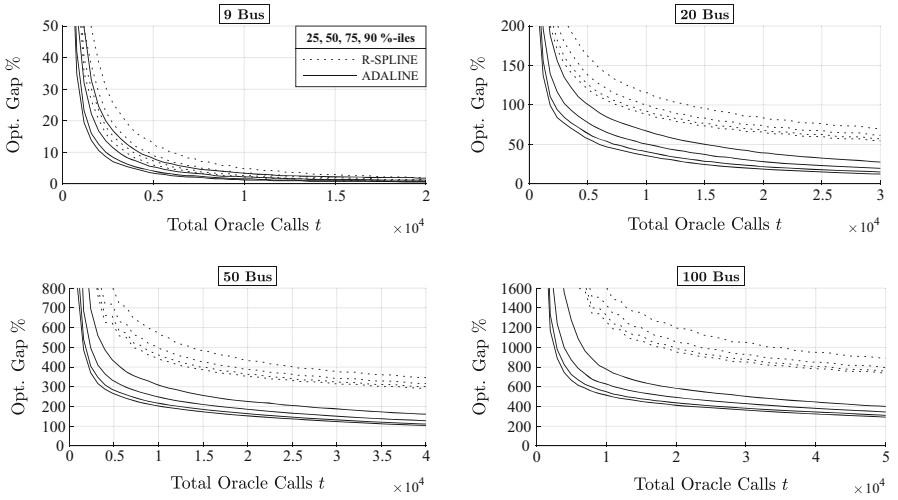

Fig. 3 Bus Scheduling: The figure shows 25th, 50th, 75th, and 90th percentiles of the optimality gap at the end of t total stochastic oracle calls, calculated across 1000 independent runs of each algorithm on the 9, 20, 50, and 100 bus scheduling problems

$c_1 = 1.1$, and the initial step length and the step size expansion factor during line search were $s_0 = 2$ and $c = 2$. In ADALINE the lower bound sample size sequence was $\lambda_k = \lceil \max(2 \log k, 2) \rceil$; the initial step size and step expansion factors during the line search were $\sqrt{d}$ and 2.0, respectively. In the kth iteration, the percent optimality gap is calculated as $100(g(X_k) - g(x^*))/g(x^*)$.

In all four plots of Fig. 3, ADALINE demonstrates superior performance relative to R- SPLINE, where ADALINE's superiority is more dramatic on higher-dimensional problems. While R- SPLINE forces sampling $2d$ neighbors after every line search, ADALINE's adaptive sampling framework allows it to begin a new line search as soon as it determines a direction that, with sufficient probability, is a descent direction. In addition, the bus scheduling problem suite is deterministically constrained, and ADALINE handles direction-finding at the constraint better than R- SPLINE.

5.2 News vendor with dynamic consumer substitution

Suppose a random number of customers labeled $1, 2, \ldots, N$ arrive in sequence to a store on any day, each wishing to purchase a particular product chosen from $d + 1$ available options that are labeled $0, 1, 2, \ldots, d$. Assume that the option labeled "0" is the "no purchase" option indicating that either the customer chooses not to buy, or that none of the products are available. The cost price and the selling price per unit of product under option $j \in \{0, 1, 2, \ldots, d\}$ are c_j and s_j, respectively, with $c_0 = s_0 = 0$. The store also incurs a unit cost p associated each product stocked at the beginning of the day.

Arriving customers are assumed to make their product choice through an implicit utility maximization process. Formally, suppose the vector $x(i) = (x_1(i), x_2(i), \ldots,$

$x_d(i)) \in [0, \infty)^d$ represents the stocking level when the i-th customer arrives at the store, and that the set

$$\mathcal{S}_i := \left\{ j \in \{1, 2, \ldots, d\} : x_j(i) > 0 \right\} \cup \{0\}$$

represents the choices available to this customer. The i-th customer then assigns an (unknown) utility $U_j(i)$, $j \in \mathcal{S}_i$ to each available option, and then selects that product D_i having the highest utility:

$$D_i := \arg\max_{j \in \mathcal{S}_i} U_j(i).$$

The choice vector $(D_1, D_2, \ldots, D_N)$ thus represents a "sample-path" corresponding to the store's operation on a day, has a distribution that is unknown in closed-form but from which observations can be simulated.

Under the above setting, we wish to identify the stocking levels $x = (x_1, x_2, \ldots, x_d) \in [0, \infty)^d$ that maximize the store's expected profit each day, that is, we wish to solve the optimization problem:

$$\underset{x \in [0,\infty)^d}{\text{maximize}} \quad \mathbb{E}\left[\sum_{i=1}^{N} (s_{D_i} - c_{D_i}) D_i \right] - p \times \sum_{j=1}^{d} x_j, \tag{26}$$

where the expectation "integrates out" the randomness due to the number of customers N and the randomness in their choices.

The optimization problem just outlined, a variation of which appears in [32], is not solvable in closed-form. The main challenge to algorithmic solutions is the obvious and severe combinatorial nature of the problem. For testing the performance of ADALINE on (26), we deliberately choose problem settings for which a closed-form solution can be calculated. For example, let the random variable N be discrete uniform over $\{a, a+1, a+2, \ldots, b \times d\}$ where $a = 10$ and $b = 5$, and the number of products d will be varied to yield different optimization problems of the kind (26). Let $U_j(i) = V_j(i) + \xi_j(i)$, where $V_j(i) = 1 + j$, $j = 1, 2, \ldots, d$ and $V_0(i) = -\infty$ for each $i = 1, 2, \ldots, N$, and $\xi_j(i)$, $j = 1, 2, \ldots, d$, $i = 1, 2, \ldots, N$ are i.i.d. Gumbel [19] random variables, implying that for all $i = 1, 2, \ldots, N$:

$$P(D_i = 0) = 0; \quad \text{and} \quad P(D_i = j) = \frac{\exp\{V_j\}}{\sum_{j=1}^{d} \exp\{V_j\}}, \quad j = 1, 2, \ldots, d.$$

Finally, suppose we choose $s_i, c_i, i = 1, 2, \ldots, d$ so that $\text{argmax}\{s_j - c_j : j = 1, 2, \ldots, d\}$ is unique, that is, the option yielding the highest unit profit is unique. With these choices, we can demonstrate that the optimal solution to (26) is to stock only the option yielding the largest unit profit, that is, $x_j^* = 0$ if $j \neq \text{argmax}\{s_j - c_j : j = 1, 2, \ldots, d\}$, and that $x_j^* = \lfloor b - (p/(s_j - c_j)) \times (b - a + 1) \rfloor$ if $j = \text{argmax}\{s_j - c_j : j = 1, 2, \ldots, d\}$.

Figure 4 illustrates the performance of ADALINE and R- SPLINE when solving (26) for $d = 5, 15, 25, 50$. Figure 4 displays the 25th, 50th, 75th and 90th

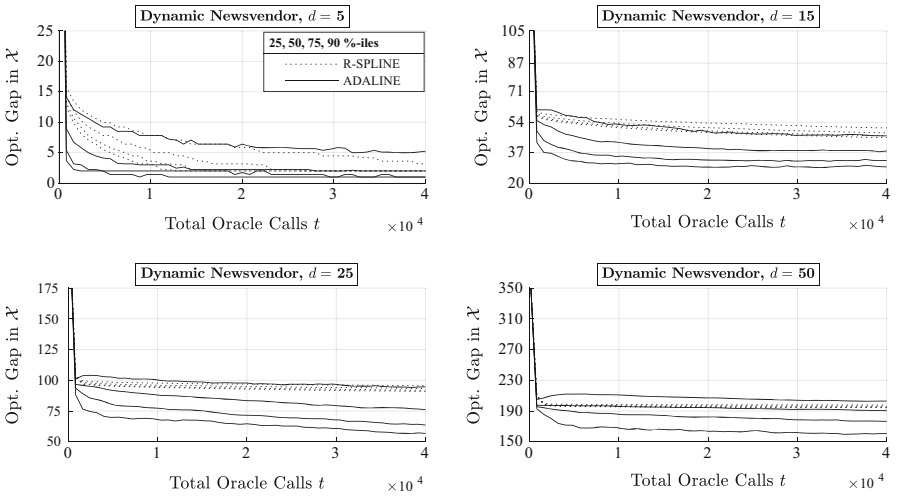

Fig. 4 Newsvendor with Dynamic Consumer Substitution: The figure shows 25th, 50th, 75th, and 90th percentiles of the optimality gap, measured in the decision space, at the end of t total stochastic oracle calls, calculated across 1000 independent runs of each algorithm on the 5, 15, 25, and 50 dimension newsvendor problems

percentiles of the optimality gap measured as the L_2 norm of the difference $\|X_t - x^*\|$ between the solution X_t returned by the algorithm after expending t units of computational effort and the optimal solution x^*. We use this metric instead of the usual optimality gap $f(X_t) - f(x^*)$ because this quantity is unknown in closed form, and estimating it with any accuracy would require enormous computational effort.

Both algorithms ADALINE and R- SPLINE reliably solve problems within this problem suite, rapidly approaching the vicinity of the optimal solution and then exhibiting a trajectory that resembles a random walk. ADALINE predictably dominates R- SPLINE except in the 90th percentile, where the performances of ADALINE and R- SPLINE are comparable. ADALINE's dominance as the dimension d becomes larger should be no surprise considering ADALINE's local optimality test, which is especially efficient relative to R- SPLINE when sampling in a high-dimensional neighborhood. As an example, when $d = 25$, the number of vertices in any unit cube containing the optimal solution is approximately $2^{25} \approx 10^{1.5} \times 10^6$, rendering effective local searching especially important for efficiency.

Another difference between ADALINE and R- SPLINE that becomes evident during implementation is the effect of CRN. When CRN is effective, e.g., when the variance of N is negligible, R- SPLINE tends to be very effective and difficult to beat. This is because R- SPLINE is designed to exploit the structure in sample-paths, yielding trajectories that look like those obtained from the execution of an algorithm solving a deterministic problem. As the variance of N grows, however, CRN becomes less effective, allowing the statistical local search features of ADALINE to become more important.

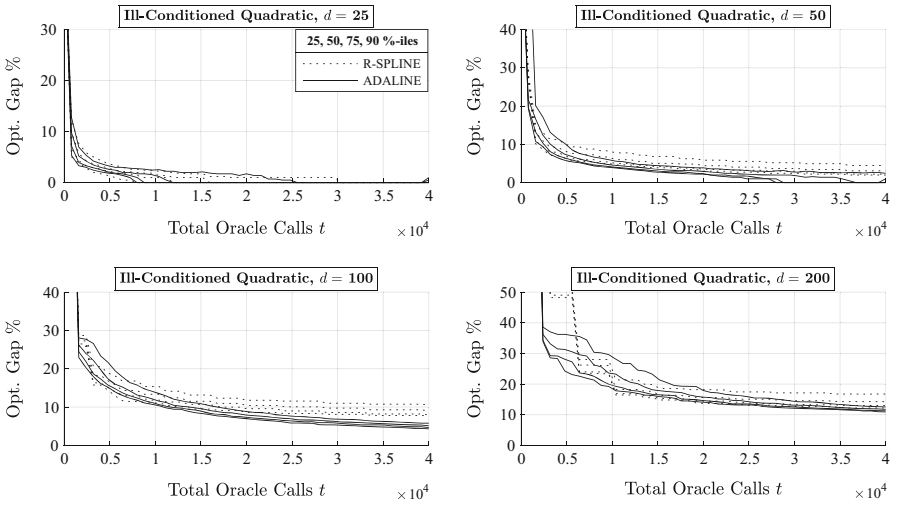

Fig. 5 Ill-Conditioned Quadratic: The figure shows 25th, 50th, 75th, and 90th percentiles of the optimality gap at the end of t total stochastic oracle calls, calculated across 60 independent runs of each algorithm on the 25, 50, 100, and 200 dimension quadratic problems

5.3 Ill-conditioned discrete quadratic problem suite

We consider the d-dimensional ill-conditioned discrete quadratic problem

$$\text{minimize}_{x \in \mathbb{Z}^d} \quad f(x) := \mathbb{E}[F(x, \xi)] = x^\intercal B x \tag{27}$$

where B is a $d \times d$, symmetric, positive-definite matrix, and $F(x, \xi) = x^\intercal B x + \xi_1(x) + \xi_2$. In the expression for $F(x, \xi)$, $\xi_1(x)$ is a real-valued normal random variable with mean zero and standard deviation $\sigma_1 \times \|Bx\|$, where $\sigma_1 = 5$; ξ_2 is a normal random variable with mean zero and standard deviation $\sigma_2 = 5$.

The problem is designed to be challenging in three ways. First, the variance of the objective function estimates are proportional to the squared norm of the function gradient, implying that sample sizes should be chosen judiciously. Second, the matrix B can be ill-conditioned in the sense of having large condition number κ, especially when the dimension d is large. Third the sample path problems are elliptically symmetric with a possible local maximum at zero, depending on the random variables $\xi_1(x)$ and ξ_2.

We apply ADALINE and R-SPLINE on the above problem with the same algorithm parameter values as in the bus scheduling problem and with the initial solution $x^* = (10, \cdots, 10)$. Figure 5 illustrates the performance of the algorithms, where the optimality gap is calculated as $\|X_k\|$.

Both ADALINE and R-SPLINE exhibit good performance on this problem, with a rapid initial approach to the vicinity of the solution at the origin. The performance of both algorithms are comparable during the low to moderate amounts of oracle effort, after which ADALINE seems to perform better. This is again due to the superior Proec-

dure NE within ADALINE which allows termination soon after a descent direction is identified with a high probability. No such mechanism exists in R-SPLINE, where at least $d+1$ points are observed during each call of the neighborhood enumeration step. For the same reason, one can observe that R-SPLINE's trajectory is more step-like, and is especially pronounced for $d = 200$.

Acknowledgements The third author fondly remembers his personal and research interactions with Shabbir Ahmed. Shabbir was an amazing scholar who made fundamental contributions to stochastic programming. The third author also thanks A. Villukanti and S. Venkatramanan at the Biocomplexity Institute, University of Virginia for discussions that led to the incorporation of some ideas within ADALINE, and its name.

A Procedure LI: line search

Procedure LI, listed in Algorithm 3, is straightforward. Starting with the candidate next iterate $\widetilde{X}_k$, LI successively observes objective estimates obtained with sample size M_k at points that are "closest" to the line $\widetilde{X}_k + t\hat{d}_k$, $t \in \mathbb{R}$, as long as the observed objective estimates are monotone decreasing. More precisely, given the starting point $W_0 \coloneqq \widetilde{X}_k$ and direction $\hat{d}_k$, LI obtains objective estimates at points $W_\ell \coloneqq \operatorname{argmin}\{\|x - (\widetilde{X}_k + 2^{\ell-1} s_0 \hat{d}_k)\| : x \in \mathcal{X} \setminus \{\widetilde{X}_k\}\}$, $\ell = 1, 2, 3, \ldots$ where s_0 is a fixed constant that defaults to $s_0 = 1$. The argmin operation is computationally trivial since the neighbors of the point $\widetilde{X}_k + t\hat{d}_k$ can be obtained by rounding. The line search proceeds as long as the sequence $\bar{F}(W_\ell, M_k)$, $\ell = 0, 1, 2, \ldots$ is non-increasing, or a pre-specified limit on the maximum number of steps in the line search is reached. Finally, Procedure LI performs a simple bisection search to find a better point between the penultimate point and the last point, as the last step size may be large.

Algorithm 3: $\widetilde{X}_k = \mathrm{LI}(\widetilde{X}_k, \hat{d}_k, M_k, b_1)$

Input: candidate next iterate $\widetilde{X}_k$; direction $\hat{d}_k$; sample size M_k; limit b_1
Output: updated candidate next iterate $\widetilde{X}_k$

Initialize: $s_0 \leftarrow 1$, $\ell \leftarrow 0$, $W_\ell \leftarrow \widetilde{X}_k$, and $\bar{F}(W_\ell, M_k) \leftarrow \bar{F}(\widetilde{X}_k, M_k)$
repeat `/begin monotone decreasing search`
  $\ell \leftarrow \ell + 1$
  $W_\ell \leftarrow \operatorname{argmin}\{\|x - (\widetilde{X}_k + 2^{\ell-1} s_0 \hat{d}_k)\| : x \in \mathcal{X} \setminus \{\widetilde{X}_k\}\}$, observe $\bar{F}(W_\ell, M_k)$
until $\bar{F}(W_\ell, M_k) > \bar{F}(W_{\ell-1}, M_k)$ *or* $\ell \geq \lfloor b_1/2 \rfloor$
Initialize: $W_{\text{low}} \leftarrow W_{\ell-1}$, $W_{\text{high}} \leftarrow W_\ell$ `/begin bisection`
while $\|W_{\text{low}} - W_{\text{high}}\| > \sqrt{d}$ **do**
  set $W_{\text{mid}} \leftarrow \operatorname{argmin}\{\|x - (W_{\text{low}} + W_{\text{high}})/2\| : x \in \mathcal{X}\}$, observe $\bar{F}(W_{\text{mid}}, M_k)$
  if $\bar{F}(W_{\text{mid}}, M_k) < \bar{F}(W_{\text{low}}, M_k)$ **then** $W_{\text{high}} \leftarrow W_{\text{low}}$, $W_{\text{low}} \leftarrow W_{\text{mid}}$
  else $W_{\text{high}} \leftarrow W_{\text{mid}}$
return $\widetilde{X}_k \leftarrow W_{\text{low}}$ `/accept point`

B Procedure DA: estimating a descent direction

Procedure DA estimates a descent direction $\hat{d}_k$ at the candidate iterate $\widetilde{X}_k$ or finds that $\widetilde{X}_k$ is an estimated local minimizer in its $\mathcal{N}_1$-neighborhood. DA performs the following steps:

1. DA enumerates the neighbors of $\widetilde{X}_k$ looking for (a) at least $d+1$ feasible neighbors that form a simplex with volume in d dimensions, and (b) at least one better neighbor. (See Algorithm 4 steps 1–14.)
2. If there are no better neighbors, DA returns with $\mathcal{B}$ is true. (See Algorithm 4 step 15.)
3. Otherwise, DA constructs an estimated descent cone and estimated descent direction. (See Algorithm 4 steps 17–21.)

Finally, if DA identifies an estimated better neighbor, it updates the candidate next iterate. As our analysis holds with or without this "hop," for simplicity, we omit it from Sect. 4.

Algorithm 4: $[\hat{d}_k, \widetilde{X}_k, \mathfrak{B}] = \mathrm{DA}(\widetilde{X}_k, M_k, \lambda_k, \alpha, \delta)$

Input: candidate iterate, $\widetilde{X}_k$; incumbent sample size, $\widetilde{M}_k$; minimum sample size, λ_k; significance level, $\alpha \in (0, 1)$; number of observations δ to obtain at once.
Output: estimated descent direction $\hat{d}_k$ and Boolean $\mathfrak{B}$.

1 Generate $Y = (y_1, \ldots, y_d)$ where each y_j is a random variate uniform on $\{-1, 1\}$.
2 Initialize: $\mathcal{V} \leftarrow \emptyset$, $\mathfrak{B} \leftarrow 1$ `/first pass through d neighbors`
3 **for** $j = 1, \ldots, d$ **do**
4 $v_j \leftarrow \widetilde{X}_k + y_j e_j$ where e_j is a vector of zeros with 1 in the jth place
5 **if** v_j *is feasible* **then**
6 $\mathcal{V} \leftarrow \mathcal{V} \cup \{v_j\}$; **if** $\bar{F}(v_j, M_k) < \bar{F}(\widetilde{X}_k, M_k)$ **then** $\mathfrak{B} \leftarrow 0$
7 Initialize: $j \leftarrow 0$, $c \leftarrow 0$ `/second pass on "other side"`
8 **while** *(*$\mathfrak{B} = 1$ *or* $|\mathcal{V}| < d + 1$*) and* $j < d$ **do**
9 $j \leftarrow j + 1$ and $\tilde{v}_j \leftarrow \widetilde{X}_k - y_j e_j$
10 **if** $\tilde{v}_j$ *is feasible* **then**
11 **if** $v_j \notin \mathcal{V}$ *or* $c = 0$ **then** `/if` $v_j \notin \mathcal{V}$ `or first feasible then add` $\tilde{v}_j$
12 $\mathcal{V} \leftarrow \mathcal{V} \cup \{\tilde{v}_j\}$, $c \leftarrow 1$; **if** $\bar{F}(\tilde{v}_j, M_k) < \bar{F}(\widetilde{X}_k, M_k)$ **then** $\mathfrak{B} \leftarrow 0$
13 **else if** $\mathfrak{B} = 1$ *and* $\bar{F}(\tilde{v}_j, M_k) < \bar{F}(\widetilde{X}_k, M_k)$ **then**
14 $\mathcal{V} \leftarrow (\mathcal{V} \setminus \{v_j\}) \cup \{\tilde{v}_j\}$ and $\mathfrak{B} \leftarrow 0$ `/replace` v_j `with` $\tilde{v}_j$
15 **if** $\mathfrak{B} = 1$ **then**
16 **return** $\hat{d}_k \leftarrow 0, \widetilde{X}_k, \mathfrak{B}$ `/return: no better neighbors`
17 **else** `/construct estimated descent cone and direction`
18 $\mathcal{C}(\widetilde{X}_k) \coloneqq \{v_j - \widetilde{X}_k : \bar{F}(v_j, M_k) < \bar{F}(\widetilde{X}_k, M_k), v_j \in \mathcal{V}\}$
19 $\hat{d}_k \leftarrow \sum_{\hat{d} \in \mathcal{C}(\widetilde{X}_k)} T_k(\hat{d} + X_k) \times \hat{d}$ where $T_k(\cdot)$ is from (6)
20 $v_* \leftarrow \mathrm{argmin}\{\bar{F}(v, \widetilde{M}_k) : v \in \mathcal{V}\}$ `/identify better neighbor`
21 **return** $\hat{d}_k, \widetilde{X}_k \leftarrow v_*, \mathfrak{B}$ `/return: direction found`

References

1. Abhishek, K., Leyffer, S., Linderoth, J.: A mixed-integer nonlinear program for the optimization of thermal insulation systems. Opt. Eng. **11**(2), 185–212 (2010)
2. Abramson, M., Audet, C., Chrissis, J., Walston, J.: Mesh adaptive direct search algorithms for mixed variable optimization. Opt. Letters **3**(1), 35–47 (2009)
3. Asmussen, S., Glynn, P.W.: Stochastic Simulation: Algorithms and Analysis, Stochastic modeling and applied probability, vol. 57. Springer, New York (2007)
4. Audet, C., Dennis Jr., J.: Pattern search algorithms for mixed variable programming. SIAM J. Opt. **11**(3), 573–594 (2000)
5. Billingsley, P.: Probability and Measure, 3rd edn. Wiley, New York (1995)
6. Bonami, P., Biegler, L., Conn, A., Cornuéjols, G., Grossmann, I., Laird, C., Lee, J., Lodi, A., Margot, F., Sawaya, N., Wächter, A.: An algorithmic framework for convex mixed integer nonlinear programs. Discrete Opt. **5**(2), 186–204 (2008)
7. Bottou, L., Curtis, F.E., Nocedal, J.: Optimization methods for large-scale machine learning. SIAM Rev. **60**(2), 223–311 (2018). https://doi.org/10.1137/16M1080173
8. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, New York (2004)
9. Chen, C.H., Lin, J., Yücesan, E., Chick, S.E.: Simulation budget allocation for further enhancing the efficiency of ordinal optimization. Dis. Event Dyn. Syst. **10**(3), 251–270 (2000). https://doi.org/10.1023/A:1008349927281
10. Eubank, S., Guclu, H., Kumar, V.S.A., Marathe, M.V., Srinivasan, A., Toroczkai, Z., Wang, N.: Modelling disease outbreaks in realistic urban social networks. Nature **429**, 180–184 (2004)
11. Gosavi, A., Ozkaya, E., Kahraman, A.F.: Simulation optimization for revenue management of airlines with cancellations and overbooking. OR Spect. **29**(1), 21–38 (2007). https://doi.org/10.1007/s00291-005-0018-z
12. Hashemi, F., Ghosh, S., Pasupathy, R.: On adaptive sampling rules for stochastic recursions. In: A. Tolk, S.Y. Diallo, I.O. Ryzhov, L. Yilmaz, S. Buckley, J.A. Miller (eds.) Proceedings of the 2014 Winter Simulation Conference, pp. 3959–3970. IEEE, Piscataway, NJ (2014). https://doi.org/10.1109/WSC.2014.7020221
13. Henderson, S.G., Pasupathy, R.: Simulation optimization library (2021). https://github.com/simopt-admin/simopt/wiki
14. Hong, L.J., Nelson, B.L.: Discrete optimization via simulation using COMPASS. Oper. Res. **54**(1), 115–129 (2006). https://doi.org/10.1287/opre.1050.0237
15. Hu, L., Andradottir, S.: An asymptotically optimal set approach for simulation optimization. INFORMS J. Comput. (2018). https://doi.org/10.1287/ijoc.2018.0811
16. Hunter, S.R., Nelson, B.L.: Parallel ranking and selection. In: A. Tolk, J. Fowler, G. Shao, E. Yücesan (eds.) Advances in Modeling and Simulation: Seminal Research from 50 Years of Winter Simulation Conferences, Simulation Foundations, Methods and Applications, chap. 12, pp. 249–275. Springer International, Switzerland (2017). https://doi.org/10.1007/978-3-319-64182-9
17. Jalali, H., Van Nieuwenhuyse, I.: Simulation optimization in inventory replenishment: a classification. IIE Trans. **47**(11), 1217–1235 (2015). https://doi.org/10.1080/0740817X.2015.1019162
18. Johnson, N.L., Kotz, S., Balakrishnan, N.: Continuous Multivariate Distributions. Wiley, New York (2000)
19. Karatzas, I., Yor, M., Embrechts, P.: Modelling Extremal Events: for Insurance and Finance, vol. 33. Springer, Berlin (1997)
20. Kim, S., Pasupathy, R., Henderson, S.G.: A guide to SAA. In: M. Fu (ed.) Encyclopedia of Operations Research and Management Science, Hillier and Lieberman OR Series. Elsevier (2014)
21. Kim, S.H., Nelson, B.L.: Selecting the best system. In: Henderson, S.G., Nelson, B.L. (eds.) Simulation, Handbooks in Operations Research and Management Science, vol. 13, pp. 501–534. Elsevier, Amsterdam (2006)
22. Kleywegt, A.J., Shapiro, A., Homem-de-Mello, T.: The sample average approximation method for stochastic discrete optimization. SIAM J. Opt. **12**, 479–502 (2001)
23. Kushner, H., Yin, G.G.: Stochastic approximation and recursive algorithms and applications. Stochastic Model. Appl. Prob. (2003). https://doi.org/10.1007/b97441
24. Lamiri, M., Grimaud, F., Xie, X.: Optimization methods for a stochastic surgery planning problem. Int. J. Prod. Econ. **120**(2), 400–410 (2009). https://doi.org/10.1016/j.ijpe.2008.11.021
25. Law, A.M.: Simulation Modeling and Analysis, 5th edn. McGraw Hill Education, New York (2015)

26. Le Digabel, S., Wild, S.M.: A taxonomy of constraints in simulation-based optimization. arXiv (2015). arxiv:1505.07881
27. Lee, S., Nelson, B.L.: General-purpose ranking and selection for computer simulation. IIE Trans **48**(6), 555–564 (2016)
28. Li, P., Abbas, M., Pasupathy, R., Head, L.: Simulation-based optimization of maximum green setting under retrospective approximation framework. Transport. Res. **2192**, 1–10 (2010). https://doi.org/10.3141/2192-01
29. Liuzzi, G., Lucidi, S., Rinaldi, F.: Derivative-free methods for bound constrained mixed-integer optimization. Dis. Opt. **53**(2), 505–526 (2011)
30. Liuzzi, G., Lucidi, S., Rinaldi, F.: Derivative-free methods for mixed-integer constrained optimization problems. J. Opt. Theory Appl. **164**(3), 933–965 (2015)
31. Lorenz, A.J., Chao, S., Asoro, F.G., Heffner, E.L., Hayashi, T., Iwata, H., Smith, K.P., Sorrells, M.E., Jannink, J.L.: Genomic selection in plant breeding: Knowledge and prospects. In: Sparks, D.L. (ed.) Advances in Agronomy, vol. 110, pp. 77–123. Academic Press, Elsevier, San Diego, CA (2011)
32. Mahajan, S., van Ryzin, G.: Stocking retail assortments under dynamic consumer substitution. Oper. Res. **49**(3), 334–351 (2001)
33. Murota, K.: Discrete convex analysis. SIAM, Philadelphia (2003)
34. Nesterov, Y.: Introductory lectures on convex optimization: A basic course, *Applied Optimization*, vol. 87. Springer (2004). https://doi.org/10.1007/978-1-4419-8853-9
35. Newton, D., Yousefian, F., Pasupathy, R.: Stochastic Gradient Descent: Recent Trends, chap. 9, pp. 193–220. INFORMS (2018). https://doi.org/10.1287/educ.2018.0191
36. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer Series in Operations Research and Financial Engineering. Springer, New York (2006)
37. Pasupathy, R., Ghosh, S.: Simulation optimization: a concise overview and implementation guide. In: H. Topaloglu (ed.) TutORials in Operations Research, chap. 7, pp. 122–150. INFORMS, Catonsville, MD (2013). https://doi.org/10.1287/educ.2013.0118
38. Pasupathy, R., Henderson, S.G.: A testbed of simulation-optimization problems. In: L.F. Perrone, F.P. Wieland, J. Liu, B.G. Lawson, D.M. Nicol, R.M. Fujimoto (eds.) Proceedings of the 2006 Winter Simulation Conference, pp. 255–263. IEEE, Piscataway, NJ (2006). https://doi.org/10.1109/WSC.2006.323081
39. Pasupathy, R., Henderson, S.G.: SimOpt: A library of simulation optimization problems. In: S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, M. Fu (eds.) Proceedings of the 2011 Winter Simulation Conference, pp. 4075–4085. IEEE, Piscataway, NJ (2011). https://doi.org/10.1109/WSC.2011.6148097
40. Pasupathy, R., Schmeiser, B.W.: Root finding via darts: Dynamic adaptive random target shooting. In: B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, E. Yücesan (eds.) Proceedings of the 2010 Winter Simulation Conference, pp. 1255–1262. Institute of Electrical and Electronics Engineers, Inc., Piscataway, NJ (2010). http://www.informs-sim.org/wsc10papers/115.pdf
41. Pasupathy, R., Hunter, S.R., Pujowidianto, N.A., Lee, L.H., Chen, C.: Stochastically constrained ranking and selection via SCORE. ACM Trans.Model. Comput. Sim. **25**(1), 1–26 (2015). https://doi.org/10.1145/2630066
42. Pasupathy, R., Glynn, P.W., Ghosh, S., Hashemi, F.: On sampling rates in simulation-based recursions. SIAM J. Opt. **28**(1), 45–73 (2018). https://doi.org/10.1137/140951679
43. Robbins, H., Siegmund, D.: A convergence theorem for non negative almost super-martingales and some applications. In: J.S. Rustagi (ed.) Optimizing Methods in Statistics, pp. 233–257. Academic Press (1971). https://doi.org/10.1016/B978-0-12-604550-5.50015-8
44. Salemi, P.L., Song, E., Nelson, B.L., Staum, J.: Gaussian Markov random fields for discrete optimization via simulation: Framework and algorithms. Oper. Res. **67**(1), 250–266 (2019). https://doi.org/10.1287/opre.2018.1778
45. Shapiro, A., Dentcheva, D., Ruszczyński, A.: Lectures on Stochastic Programming: Modeling and Theory. MPS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, Philadelphia, PA (2009)
46. Shashaani, S., Hashemi, F.S., Pasupathy, R.: ASTRO-DF: A class of adaptive sampling trust-region algorithms for derivative-free simulation optimization. SIAM J. Opt. **28**(4), 3145–3176 (2018)
47. Shin, D., Broadie, M., Zeevi, A.: Tractable sampling strategies for ordinal optimization. Oper. Res. **66**(6), 1693–1712 (2018). https://doi.org/10.1287/opre.2018.1753

48. Sun, L., Hong, L.J., Hu, Z.: Balancing exploitation and exploration in discrete optimization via simulation through a gaussian process-based search. Oper. Res. **62**(6), 1416–1438 (2014). https://doi.org/10.1287/opre.2014.1315
49. Vershynin, R.: High-Dimensional Probability: An Introduction with Applications in Data Science, Cambridge Series in Statistical and Probabilistic Mathematics, vol. 47. Cambridge University Press, Cambridge, UK (2018)
50. Wang, H., Pasupathy, R., Schmeiser, B.W.: Integer-ordered simulation optimization using R-SPLINE: Retrospective Search using Piecewise-Linear Interpolation and Neighborhood Enumeration. ACM Transactions on Modeling and Computer Simulation **23**(3), 17:1–17:24 (2013). https://doi.org/10.1145/2499913.2499916
51. Williams, D.: Probability with martingales. Cambridge University Press, Cambridge, UK (1991)
52. Xu, J., Nelson, B.L., Hong, L.J.: Industrial Strength COMPASS: A comprehensive algorithm and software for optimization via simulation. ACM Trans.Model. Comput. Simul. **20**, 1–29 (2010)
53. Zhang, H., Zheng, Z., Lavaei, J.: Discrete convex simulation optimization (2020)
54. Zhang, H., Zheng, Z., Lavaei, J.: Stochastic localization methods for discrete convex simulation optimization (2020)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Mathematical Programming is a copyright of Springer, 2021. All Rights Reserved.