Operations Research Letters 50 (2022) 347–355

Contents lists available at ScienceDirect

Operations Research Letters

www.elsevier.com/locate/orl

Iteratively sampling scheme for stochastic optimization with variable number sample path

Shuang Hao, Dali Zhang*, Ming Dong

ShenZhen Research Institute, & Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai, 200030, China

ARTICLE INFO

Article history:
Received 17 December 2020
Received in revised form 28 March 2022
Accepted 29 March 2022
Available online 12 April 2022

Keywords:
Sampling scheme
Variable number sample path
Directional direct search
Convergence

ABSTRACT

Optimal search methods are proposed for solving optimization problems with analytically unobtainable objectives. This paper proposes a method by incorporating sampling schemes into the directional direct search with variable number sample path and investigates its effectiveness in solving stochastic optimization problems. We also explore the conditions on sample sizes at each iteration under which the convergence in probability can be guaranteed. Finally, a set of benchmark problems are numerically tested to show the effectiveness in different sampling schemes.

1. Introduction

Consider unconstrained stochastic optimization problem

$$\min_{x\in\mathbb{R}^p}\left\{F(x) := \mathbb{E}\left[f\left(x,\xi(\omega)\right)\right]\right\} \tag{1}$$

where $\xi : \Omega \to \Xi \subset \mathbb{R}^q$ is a random vector defined on a probability space $(\Omega, \mathcal{F}, P)$, $\mathbb{E}[\cdot]$ is the mathematical expectation operator. Throughout this paper, we assume that $\mathbb{E}[f(x,\xi(\omega))]$ is well defined for every $x \in \mathbb{R}^p$ and the distributions of ξ cannot be analytically obtained. To ease the notation, we write $\xi(\omega)$ as ξ and the context will make it clear when ξ should be interpreted as a deterministic vector.

One standard technique for solving the problem (1) is the sample average approximation (SAA) approach. By incorporating an independently identically distributed (iid) sample $\boldsymbol{\xi}^N := (\xi_1, \cdots, \xi_N)^\top \in \Xi^N \subset \mathbb{R}^q \times \cdots \times \mathbb{R}^q$, this method attains a "deterministic" approximation of the *true problem* (1) by solving

$$\min_{x\in X} F^N(x) := \frac{1}{N}\sum_{i=1}^{N} f(x,\xi_i) \tag{2}$$

where we refer (2) as *the SAA problem* with sample size N. We write $F(x,\boldsymbol{\xi}^N)$ to emphasis the randomness in $F^N(x)$ when $\boldsymbol{\xi}^N$ is not realized.

Several optimal search methods have been proposed for solving deterministic optimization problems, among which *the directional direct search (DDS)* is one of well-investigated [11,13] and proved to be effective for the unconstrained optimization problem

$$\min_{x\in\mathbb{R}^p} F(x). \tag{3}$$

The essence of the DDS is finding the optimal solution via an updating scheme for candidate solutions. The direct incorporation of the DDS for solving (2) might attain an approximation solution of (1). However, finite samples prohibit the convergence of the SAA problem. Therefore appropriate sampling schemes are necessary to guarantee the probability convergence when solving stochastic optimization problems [6]. This paper investigates sampling schemes incorporated into the DDS algorithm to ensure the convergence of the algorithm to an optimal solution of the true problem (1) and balance the effectiveness of sampling and the efficiency of the algorithm iterations.

The rest of the paper is organized as follows. Section 2 briefly reviews the related literature. In Section 3, we propose the directional direct search algorithm for unconstrained stochastic optimization and the sampling scheme to update samples iteratively. In Section 4, we analyze the convergence of the proposed algorithm. We compare the performance among different sample size schemes and perform computational study in Section 5.

2. Literature review

When incorporating SAA methods into the DDS framework, how to determine a proper sample size in each iteration remains

* Corresponding author.
E-mail addresses: sjtu-haos@sjtu.edu.cn (S. Hao), zhangdl@sjtu.edu.cn (D. Zhang), mdong@sjtu.edu.cn (M. Dong).

https://doi.org/10.1016/j.orl.2022.03.006
0167-6377/© 2022 Elsevier B.V. All rights reserved.

a main difficulty. In this study, we consider sequential sampling schemes proposed in [9] in which the samples are updated at every iteration in the algorithms.

One stream of sequential sampling approach is solving a sequence of SAA problems with increasing sample size [4,15]. At every iteration, the solution attained at preceding iteration can be used as a start point and a new SAA problem is constructed and solved. Pasupathy [14] analyzes the asymptotically optimal sample size and the error-tolerance level in the root finding problem. More recently, Royset [16] shows that this optimal convergence can rarely be realized in finite sample simulation. For stochastic programming solved by standard nonlinear programming algorithm, Bayraksan and Morton [3] develop a scheme to evaluate the quality of solutions of sequential SAA problems and determine the proper sample sizes with the aim of minimizing the total computational efforts.

Another stream is using different samples at each iteration in the iterative optimization algorithms, which is called the variable number sample path (VNSP) approach. Targeting at solving a two-stage stochastic programming with recourse, Shapiro and Homem-de Mello [17] propose a sequential quadratic programming method with different sample paths, in which the value, first and second order derivatives of the objective function are estimated from SAA at each iteration. A similar approach is adapted to optimize mixed logit model in Bastin et al. [2]. Focusing on incorporating the VNSP to the trust region approach, Deng and Ferris [7] apply Powell's unconstrained optimization method by quadratic approximation to solve a series of SAA problems under the VNSP scheme and achieve the convergence of the algorithm to a stationary point. Most recent progress on the trust region method for stochastic optimization is achieved by Chen et al. [5] which propose and analyze a trust-region model-based algorithm for solving unconstrained stochastic optimization problems.

Most of the researches on VNSP use optimization methods that require derivative information. While there are many types of problems that neither the gradient nor the structure of the objective function is known or computable, for instance, simulation-optimization problems. When solving these problems, the only known information is the estimates of function values at finite candidate solutions.

To our best knowledge, the pure random search is the first optimization approach combined with VNSP [9] for solving a stochastic optimization with a finite feasible set. Homem-de Mello [9] proves the pointwise convergence of the approximation to the true objective function under some conditions on the growth rate of sample size in each iteration. Kim and Zhang [10] and Chen and Kelley [6] propose a coordinate search algorithm for unconstrained stochastic optimization and stochastic optimization with hidden constraints respectively. Chen and Kelley [6] propose a more flexible framework in which the set of search directions is increased as the algorithm progresses and a surrogate model is solved by quasi-Newton method with approximated gradients. This is also the first work that investigate the convergence properties of deterministic direct search methods with Monte Carlo simulated objective functions. By incorporating similar sampling schemes [1] explore another derivative free algorithm, StoMADS, generalized from well-known mesh adaptive direct search, to solve stochastic blackbox optimization using probabilistic estimates, and analyzed its convergence properties in probability. One essential difference between our work and above two works is on the sampling schemes. Both studies entail a more flexible sampling scheme in which different samples are used at each function evaluation within an iteration and thus can be applied for general black-box optimization problems. The approach in this study focuses on achieving the convergence results of the variable sample path algorithm and the bounds of approximation when sample size increases sufficiently large. To this end, we requires samples being fixed independently from the solutions in each iteration. The sample can only be updated when new iteration is launched.

This paper proposes a generalized directional direct search (GDDS) method which extends the studies in [6,10]. We investigate the sampling schemes incorporated into the GDDS method. Particularly, the results also reveals how the sample size and the step-length in each iteration are related to the convergence of the algorithm in probability. For details of the direct search algorithm please refer to [11–13].

3. Algorithm

The DDS is an optimal search algorithm proposed for solving the problem

$$\min_{x\in\mathbb{R}^p} \mathbb{E}\big[f(x,\xi)\big]. \tag{4}$$

With the assumption that the explicit form of objective function $\mathbb{E}[f(x,\xi)]$ is unknown and has to be approximated by taking sample average, the DDS framework is incorporated to solve (1) with an iteratively sampling scheme.

3.1. Directional direct search

We propose a GDDS algorithm for (4) and refer the readers to its deterministic version, a generating set search method, Algorithm 3.2 in [11]. The parameters in the initialization of the GDDS algorithm consist of the initial guess $x_0 \in \mathbb{R}^p$, sample size N_0, the tolerance of step-length $\Delta_{\text{tol}} > 0$, and the initial step-length $\Delta_0 > \Delta_{\text{tol}}$. In iteration k candidate x_k is chosen by finding an appropriate pair of step-length $\Delta_k > 0$ and search direction $d_k \in \mathcal{D}_k$ which are determined by an evaluation process with expansion factor $\phi_k \geq 1$ and contraction factor $\theta_k < \theta_{\max} < 1$.

We inherit the notations from [9] by representing N_k the size of the samples used at iteration k. We call $\{N_k\}_{k=1}^{\infty}$ the schedule of sample sizes associated with the algorithm under scrutiny. Let $\Xi_\infty := \Xi^{N_1} \times \Xi^{N_2} \times \cdots$. Note that a point $\xi_\infty := (\xi_1^1, \cdots, \xi_1^{N_1}, \xi_2^1, \cdots, \xi_2^{N_2}, \cdots) \in \Xi_\infty$ represents a sample-path followed along the iterations of the algorithm. In the rest part of the paper, we rewrite ξ_k^n by ξ_n for $n = 1, 2, \cdots, N_k$ and the context will make it clear in which iteration ξ_n is generated. Again, let $\tilde{P}$ being the corresponding probability distribution of ξ_∞ on Ξ_∞ generated by P for ξ on Ξ. Moreover, in iteration k, to approximate $\mathbb{E}[f(x,\xi)]$, a sample $\boldsymbol{\xi}_k := \boldsymbol{\xi}^{N_k} = \{\xi_1, \cdots, \xi_{N_k}\}$ is generated from the distribution P or some distribution P_k close to P, and $\mathbb{E}[f(x,\xi)]$ is approximated by implementing ξ_n for $n = 1, 2, \cdots, N_k$ into $F^{N_k}(x)$ in (2). It is not difficult to see that F^{N_k} depends on $\boldsymbol{\xi}_k$ and is unknown before $\boldsymbol{\xi}_k$ being realized.

Denote by $F_k(x, \boldsymbol{\xi}_k)$ the value function in iteration k before the realization of $\boldsymbol{\xi}_k$ which is defined in support set Ξ^{N_k}. We use

$$F_k(x, \boldsymbol{\xi}_k) := \frac{1}{N_k} \sum_{n=1}^{N_k} f(x, \xi_n)$$

instead of F^{N_k} to emphasize its dependency on $\boldsymbol{\xi}_k$. We also use $\nabla F_k(x, \boldsymbol{\xi}_k)$ to denote the gradient of $F_k(x, \boldsymbol{\xi}_k)$ with respect to x for $\boldsymbol{\xi}_k \in \Xi^{N_k}$.

Algorithm 1. (**Generalized Directional Direct Search (GDDS)**)

For each iteration $k = 0, 1, 2, \ldots$

- **Step 1.** Generate sample $\boldsymbol{\xi}_k := \{\xi_1, \xi_2, \ldots, \xi_{N_k}\}$.
- **Step 2.** Evaluate $F_k(x, \boldsymbol{\xi}_k)$ at current solution $x = x_k$ and candidate solutions $x = x_k + \Delta_k d_i, \forall d_i \in \mathcal{D}_k$.

```
    $\mathcal{S}$: (Successful Iteration) If there exists $d_k \in \mathcal{D}_k$ such that $F_k(x_k + \Delta_k d_k, \boldsymbol{\xi}_k) < F_k(x_k, \boldsymbol{\xi}_k) - \rho(\Delta_k)$, then:
    - Set $x_{k+1} = x_k + \Delta_k d_k$, $\Delta_{k+1} = \phi_k \Delta_k$,
    $\mathcal{U}$: (Unsuccessful Iteration) Otherwise, $F_k(x_k + \Delta_k d, \boldsymbol{\xi}_k) \geq F_k(x_k, \boldsymbol{\xi}_k) - \rho(\Delta_k)$ for all $d \in \mathcal{D}_k$, then:
    - Set $x_{k+1} = x_k$, $\Delta_{k+1} = \theta_k \Delta_k$
    - If $\Delta_{k+1} < \Delta_{\text{tol}}$, then terminate.
• Step 3. Perform the sampling scheme to determine sample size $N_{k+1}$ in next iteration; Go to Step 1.
```

It is worth explaining some details in the algorithm. First of all, in the algorithm the set of directions $\mathcal{D}_k$ is finite and varies at different iterations. The design of $\mathcal{D}_k$ has been well investigated in [11,12]. In literature $\mathcal{D}_k$ is required to contain $p' \geq p+1$ vectors in $\mathbb{R}^p$ and positively span $\mathbb{R}^p$ and can be either changed at each iteration by including additional search directions or fixed. To focus on the sampling schemes, the GDDS algorithm in this paper inherits the framework of $\mathcal{D}_k$ designing in [11]. When $\mathcal{D}_k$ positively spans $\mathbb{R}^p$, the worst-case distance between the steepest descent direction $\nu = -\nabla f(x)$ and searching direction that makes the smallest angle with ν is measured by the cosine measure [11]

$$\kappa(\mathcal{D}_k) = \min_{\nu \in \mathbb{R}^p} \max_{d \in \mathcal{D}_k} \frac{\nu^T d}{\|\nu\| \|d\|}. \tag{5}$$

Note that the algorithm requires the capability of fixing the sample and using the same sample to evaluate the current and all the candidate points in each iterations, which might not be satisfied in some complicated situations.

Besides, the nonnegative function ρ is included to set a threshold on the acceptable decrease of the objective function at a candidate solution. In our study we use the convergence analysis and numerical tests together to reveal that the fact that condition in Algorithm 3.2 [11], $\rho(t)/t \to 0$ as $t \downarrow 0$, is not sufficient to guarantee the convergence of the GDDS algorithm. New conditions are given in Section 3.2.

3.2. Sampling scheme

The sampling scheme is a critical difference between Algorithm 3.2 in [11] and our algorithm. Different stochastic optimization algorithms might require different sampling schemes to guarantee the convergence in probability. We focus on a new type of sampling scheme where the step-length Δ_k is taken into account to determine $\{N_k\}_{k=1}^{\infty}$. In the scheme, the sets of successful and unsuccessful iterations in the algorithm are denoted as $\mathcal{S}$ and $\mathcal{U}$ respectively.

Sampling scheme.

For each iteration $k = 0, 1, 2, \ldots$

- If $k \in \mathcal{S}$, $N_{k+1} = N_k$;
- If $k \in \mathcal{U}$,

$$N_{k+1} = \max\left\{N_0, \left\lceil \beta_{k+1} \frac{\log(k+1)}{\Delta_{k+1}^2} \right\rceil\right\}. \tag{6}$$

It is important to notify that the sampling scheme given in (6) increases along with the growth of iterations and the sample sizes are enforced to be greater than N_0 to guarantee sufficient samples in early iterations. In iteration $k \in \mathcal{S}$ we can keep $N_{k+1} = N_k$ unchanged, where the success in an iteration implies that the sample averaged objective function is close enough to its real counterpart and hence can be used to determine the right direction without additional samples.

4. Convergence analysis

We investigate the convergence property of the algorithm.

Assumption 1. Assume that (i) for every $x \in \mathbb{R}^p$ random function $f(x, \xi)$ is continuously differentiable at x for almost every $\xi \in \Xi$, (ii) for all $x \in \mathbb{R}^p$, $f(x, \xi)$ is dominated by an integrable function, and (iii) $\nabla f(x, \xi)$ is Lipschitz continuous with constant M at every $x \in \mathbb{R}^p$ for almost every $\xi \in \Xi$.

Under Assumption 1, $F_k(x, \boldsymbol{\xi}_k)$ is continuously differentiable and the gradient $\nabla F_k(x, \boldsymbol{\xi}_k)$ is Lipschitz continuous with constant M at all $x \in \mathbb{R}^p$ for almost every $\boldsymbol{\xi}_\infty \in \Xi_\infty$ and $k = 1, 2, \cdots$. Thus we can verify the conditions in Theorem 7.52 [18].

Assumption 2. Function $\rho : [0, +\infty) \to \mathbb{R}^+ \cup \{0\}$ in Algorithm 1 is continuous and monotonically increasing and satisfies the following conditions

(i) $\rho(t)/t \to 0$ as $t \downarrow 0$;
(ii) (strictly sufficient decrease) $\rho(t) > 0$ for any $t > 0$.

The most difference between $\rho(t)$ and its counterpart for deterministic cases in [11] is the condition (ii) where $\rho(t) = 0$ for $t > 0$ (e.g. $\rho \equiv 0$) is excluded. We will show in the numerical tests that $\{x_k\}_{k=1}^{\infty}$ does not converge when this condition is not satisfied.

Assumption 3. The expected value and the variance of $f(x, \xi)$ satisfy the following conditions

$$\bar{F} := \inf_{x \in \mathbb{R}^p} \mathbb{E}\big[f(x, \xi)\big] > -\infty, \quad \bar{\sigma}^2 := \sup_{x \in \mathbb{R}^p} \mathrm{var}\big(f(x, \xi)\big) < \infty$$

It is worth to mention that with Assumption 3 the expectation and the variance of $f(x, \xi)$ at every candidate in the algorithm are less than $\bar{F}$ and $\bar{\sigma}^2$ respectively. In addition, we define

$$\sigma_k^2 = \max\big\{\mathrm{var}\big(f(x_k, \xi)\big), \mathrm{var}\big(f(x_k + \Delta_k d, \xi)\big)\big\}$$

which is finite for any iteration $k = 0, 1, \cdots$. We start to prove the convergence from the following proposition for $k \in \mathcal{U}$.

Proposition 1. *Under Assumption 1,*

$$\|\nabla F_k(x_k, \boldsymbol{\xi}_k)\| \leq \kappa(\mathcal{D}_k)^{-1}\left(M\Delta_k + \frac{\rho(\Delta_k)}{\Delta_k}\right) \tag{7}$$

for $k \in \mathcal{U}$ and for almost every $\boldsymbol{\xi}_k \in \Xi^{N_k}$.

The result implies that $\|\nabla F_k(x_k, \boldsymbol{\xi}_k)\|$ is either $O(\Delta_k)$ or $O(\Delta_k^{-1}\rho(\Delta_k))$ for almost every $\boldsymbol{\xi}_k \in \Xi^{N_k}$ and $k = 1, 2, \cdots$. The proof is given in the appendix. Next we show there exists a subsequence of step-length produced by Algorithm 1 converging to zero. Before proceeding to the result, we need the following well-known lemma.

Lemma 1. ***(Borel-Cantelli Lemma)*** *Let (Ω, Σ, μ) be a measure space with $\mu(\Omega) < \infty$ and suppose $\{E_n\}_{n=1}^{\infty}$ is a collection of measurable sets with $E_n \subset \Sigma$ for $n = 1, 2, \cdots$ such that $\sum_{n=1}^{\infty} \mu(E_n) < \infty$. Then*

$$P\left(\limsup_{n\to\infty} E_n\right) = 0$$

where $\limsup_{n\to\infty} E_n := \cap_{n=1}^{\infty} \cup_{k \geq n}^{\infty} E_k$.

Proposition 2. *Under Assumptions 2 and 3, the sequence of step lengths produced by Algorithm 1 satisfies*

$$\tilde{P}\left(\liminf_{k\to+\infty} \Delta_k = 0\right) = 1 \tag{8}$$

with $N_k > \beta\Delta_k^{-2}\log k$ *and* $c(\Delta_k) = \beta\rho^{-1}(\Delta_k)\Delta_k$ *for some positive constant* β *sufficiently large.*

The most challenging point in proving the convergence is that the SAA objective function varies along with iterations, while it is simply unchanged in the deterministic DDS in [11]. Therefore, the proof for [11, Theorem 3.4] cannot be applied. The proof of Proposition 2 is given in the appendix.

In fact, the result in Proposition 2 implies the value of β needs to be larger than $\Delta_k^2\rho^{-2}(\Delta_k)$ for the least Δ_k before the stop of the algorithm. Now we proceed to compare the gradients of $F_k(x_k, \boldsymbol{\xi}_k)$ and $F(x_k)$. At iteration k, we let

$$\epsilon_{\Delta_k}(x_k; d) = \frac{F(x_k) - F(x_k + \Delta_k d)}{\Delta_k}$$

for all $d \in \mathcal{D}_k$. Correspondingly, we define

$$\eta_{\Delta_k}(x_k, \boldsymbol{\xi}_k; d) = \frac{F_k(x_k, \boldsymbol{\xi}_k) - F_k(x_k + \Delta_k d, \boldsymbol{\xi}_k)}{\Delta_k}.$$

Proposition 3. *Under Assumptions 1 and 2, in iteration k of Algorithm 1, for any* $\delta > 0$ *and any* $d \in \mathcal{D}_k$*,*

$$P_k\left(\left|\eta_{\Delta_k}(x_k, \boldsymbol{\xi}_k; d) - \epsilon_{\Delta_k}(x_k; d)\right| \le \delta\right) > 1 - 4\exp\left(-N_k\frac{\delta^2\Delta_k^2}{12\sigma_k^2}\right). \tag{9}$$

The proof is given in the appendix. The result in Proposition 3 suffices to show that the probability of the event that the deviation from the first order stationarity of the SAA problem to the true counterpart surpassing a threshold can be controlled by an appropriate sampling scheme.

Theorem 1. *Under Assumptions 1 and 2, for any* $\delta > 0$*,*

$$\liminf_{k\to+\infty} P_k\left(\left\|\nabla F_k(x_k, \boldsymbol{\xi}_k) - \nabla F(x_k)\right\| \le \delta\right) = 1. \tag{10}$$

Due to the Lipschitz continuity of $\nabla F_k(\cdot, \boldsymbol{\xi}_k)$ in Assumption 1 and the convergence of Δ_k in Proposition 2, we have

$$\left|\nabla F_k(x_k, \boldsymbol{\xi}_k) - \nabla F_k(x_k + \Delta_k d, \boldsymbol{\xi}_k)\right| \le M\Delta_k\|d_k\|,$$

where $M\Delta_k$ tends to zero along with iterations with probability 1. Again, based on the result in Proposition 3 for the convergence of first order stationarity of the SAA problem, we can prove the convergence of the approximate gradient obtained from SAA. Now we present the following result.

Proposition 4. *Under Assumptions 1 and 2, let* $\{N_k\}_{k=1}^{\infty}$ *with* $N_k = \beta_k\Delta_k^{-2}\log k$ *for any sequence* $\{\beta_k\}_{k\to\infty}^{\infty}$ *satisfying* $\beta_k \to \infty$ *as* $k \to \infty$*, then*

$$\tilde{P}\left(\liminf_{k\to\infty} \|\nabla F(x_k)\| = 0\right) = 1.$$

Propositions 4 shows that to guarantee the convergence to the true stationary point in probability, we need some stronger conditions on $\{\beta_k\}_{k=1}^{\infty}$ than the conditions to guarantee the convergence in probability of the step-length Δ_k. In fact the conditions on $\{\beta_k\}_{k=1}^{\infty}$ only require that β_k goes to infinity along with the growth of k but no requirement on its growth rate. Therefore, the increment of β_k in each iteration could be at a slow pace to control the sample size.

Table 1
Fixed and variable number sample paths.

FNSP	$N_k = 200$
VNSP1(i)	$N_0 = 5,\ N_k = kN_0$
VNSP1(c)	$N_0 = 5,\ N_k = kN_0$
VNSP2(i)	$N_0 = 5,\ N_k = \beta_k\log k/\Delta_k^2$ if $k-1 \in \mathcal{U}$, $N_k = N_{k-1}$ if $k-1 \in \mathcal{S}$
VNSP2(c)	$N_0 = 5,\ N_k = \beta_k\log k/\Delta_k^2$ if $k-1 \in \mathcal{U}$, $N_k = N_{k-1}$ if $k-1 \in \mathcal{S}$

5. Numerical study

Numerical tests are comprehensively carried out in this section. First of all, we test the performances of different sampling schemes under sufficient decrease, i.e., $\rho(\cdot)$ satisfying Assumption 2, and simple decrease $\rho(\cdot) \equiv 0$ for a benchmark Rosenbrock problem. Next we implement the algorithm with sampling schemes generated by Algorithm 1 and (6) for a random Watson function minimization problem with a practical dimension. In Section 5.3 the comparative analysis are conducted on the performance of GDDS algorithm with well investigated derivative free algorithm, StoMADs [1], and use the tests in CUTEst [8] to illustrate the advantages of each algorithm.

5.1. Rosenbrock problem

We perform tests on an extended two-dimensional *Rosenbrock function* which is a benchmark in evaluating optimal search methods, since its unique global minimum solution locating inside a long and narrow valley with a parabolic shape. In the test, we intentionally add the random noise to the first decision variable in $z := (z_1, z_2)^\top$. By doing so, we have the following function

$$f(z, \xi) = 100\big(z_2 - (\xi z_1)^2\big)^2 + (\xi z_1 - 1)^2$$

and the optimization problem $\min_{z\in\mathbb{R}^2} \mathbb{E}[f(z, \xi)]$. The random noise ξ is independent with z and normally distributed with mean 1 and variance 0.1^2. The optimal solution is $z^* = (0.416, 0.175)^\top$; the optimal value is $f(z^*) = 0.463$.

One fixed and five variable number sample paths are compared which are listed in the following Table 1, where '(i)' means the sample used in each iteration are different and '(c)' means that the cumulative sampling scheme is used and $\beta_k := 0.001(1 + \log^\nu k)$ for $\nu = 0.1$. With respect to the above sampling schemes, we test simple and sufficient decrease criteria separately. We set $\rho(\Delta_k) \equiv 0$, $\phi_k = 1$, and $\theta_k = 0.5$ and $\rho(\Delta_k) = 0.5\Delta_k^2$, $\phi_k = 2$, and $\theta_k = 0.5$ in simple and sufficient decrease criteria respectively. Again in all cases, $z_0 = (-1.200, 1.000)^\top$, $\Delta_{tol} = 0.001$, $\Delta_0 = 1$ and $\mathcal{D}_k$ is the set of 4 coordinate directions.

All sampling schemes are repeated 100 times under the above two settings. The tests stop once $\Delta_k \le \Delta_{tol}$ or the number of function evaluations reach 10^6. We report the average number of function evaluations in running each sampling scheme, the average and standard deviation of $\|z_K - z^*\|$ where K is the index of iteration when the algorithm is stopped.

From the above results, we first can identify that the cumulative sampling scheme significantly reduce the total number of observations used in an algorithm where the number of overall observations are decreased from 204, 811 to 2, 021 and 88, 178 and 1, 388 for VNSP1, and from 184 to 138 and 516 to 134 for VNSP2 in simple and sufficient decrease criteria respectively. On the other hand, the effectiveness of VNSP2 can also be recognized from the results in Tables 2 and 3. The VNSP2 attains a solution closest to

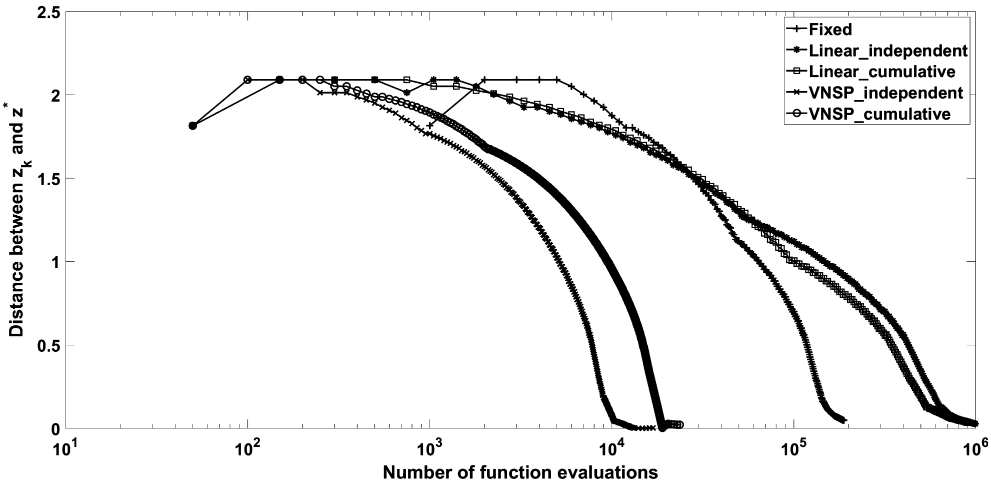

Fig. 1. Simple decrease, $\rho(\Delta_k) \equiv 0$, $\phi_k = 1$, $\theta_k = 0.5$.

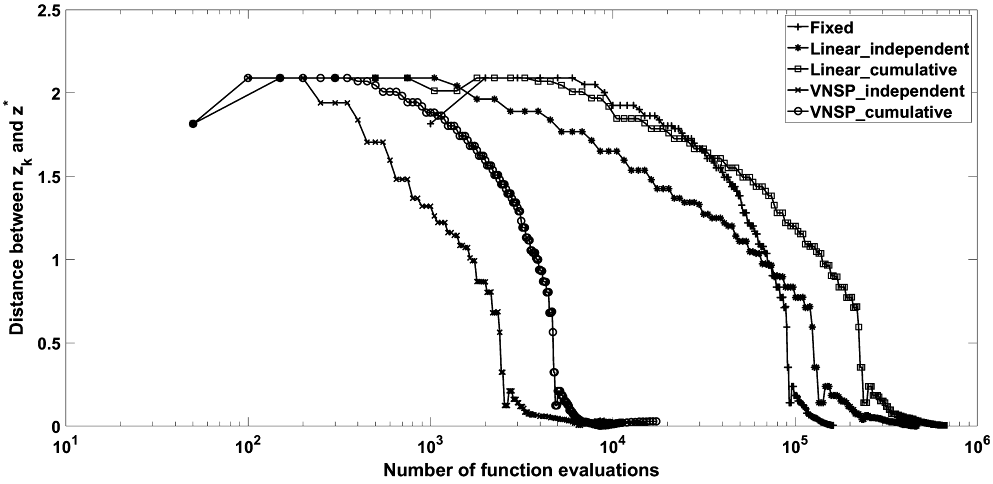

Fig. 2. Sufficient decrease, $\rho(\Delta_k) = 0.5\Delta_k^2$, $\phi_k = 2$, $\theta_k = 0.5$.

Table 2
Performances in simple decrease criterion.

	Solution Quality		Computation Cost		Sampling Cost
	avg. dist	std. dist	avg. of eval	std. of eval	sum of N_k
FNSP	0.0281	0.0121	203,550	16,594	200
VNSP1(i)	0.0271	0.0070	1,024,056	157,568	204,811
VNSP1(c)	0.0275	0.0058	1,030,777	147,155	2,021
VNSP2(i)	0.0197	0.0125	24,583	9,380	184
VNSP2(c)	0.0245	0.0135	21,898	6,977	138

Table 3
Performances in sufficient decrease criterion.

	Solution Quality		Computation Cost		Sampling Cost
	avg. dist	std. dist	avg. of eval	std. of eval	sum of N_k
FNSP	0.0147	0.0088	148,080	15,390	200
VNSP1(i)	0.0108	0.0092	440,890	161,357	88,178
VNSP1(c)	0.0128	0.0065	495,861	137,337	1,388
VNSP2(i)	0.0119	0.0095	24,621	13,499	516
VNSP2(c)	0.0184	0.0109	20,414	10,574	134

their true counterparts with the least number of function evaluations compared to the other sampling schemes in both decrease criteria. To demonstrate this, we present Figs. 1 and 2 to illustrate the computational loads versus the accuracy level achieved by different sampling schemes. It can be identified that among all the sampling schemes, the VNSP2 satisfies the conditions in Proposition 4 and achieves the best performance.

5.2. Multi-dimensional problems

In the rest part of the section, we concentrate our comparative analysis on the VNSP algorithms with independent samples at different iterations. We focus on the GDDS algorithm with VNSP2 and illustrate its capability in solving reasonably large scale problem by using *Watson functions*. *Watson function* is the sum of 31 quadratic terms of function $f_i(\cdot)$ for $i = 1, 2, \cdots, 31$ and each $f_i(\cdot)$ is a smooth function and the number of decision variables varies from 2 to 31. In this test we take into account that each variable is multiplied by a normally distributed random noise with mean 1 and variance 0.01. Consider the cases with the number of decision variables being $n = 10, 20$ and 30, and thus the random *Watson functions* are:

$$f(z,\xi) = \sum_{i=1}^{31} \big(f_i(z,\xi)\big)^2,$$

$$f_i(z,\xi) = \sum_{j=2}^{n} (j-1)\xi_j z_j t_i^{j-2} - \left(\sum_{j=1}^{n} \xi_j z_j t_i^{j-1}\right)^2 - 1,$$

$$f_{30}(z,\xi) = \xi_1 z_1, \quad f_{31}(z,\xi) = \xi_2 z_2 - (\xi_1 z_1)^2 - 1$$

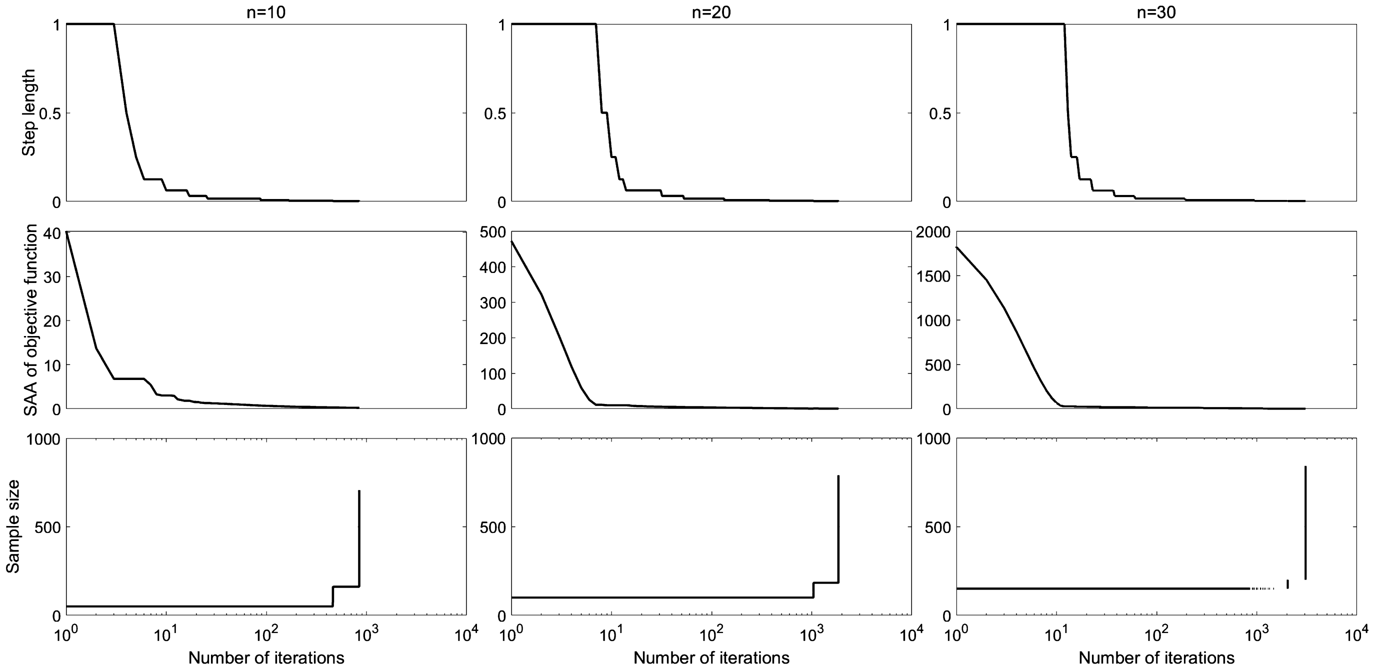

Fig. 3. Simple decrease, $\rho(\Delta_k) \equiv 0$, $\phi_k = 1$, $\theta_k = 0.5$.

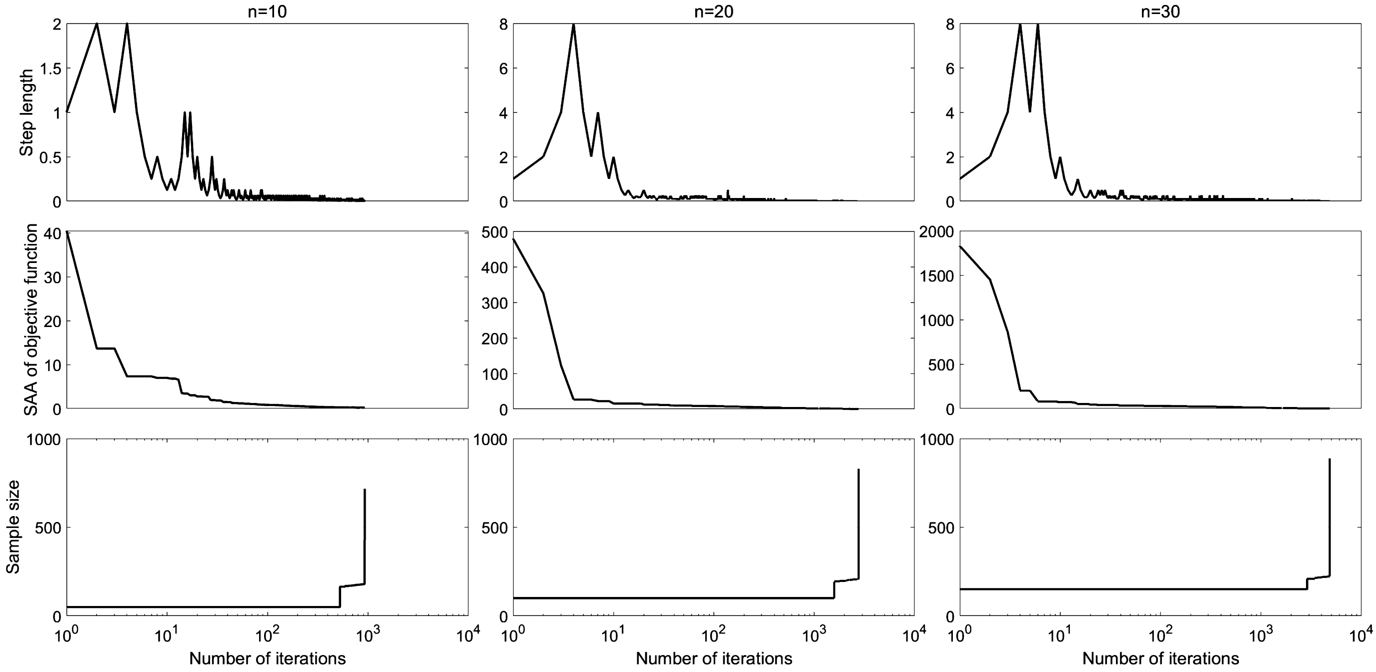

Fig. 4. Sufficient decrease, $\rho(\Delta_k) = 0.5\Delta_k^2$, $\phi_k = 2$, $\theta_k = 0.5$.

where $t_i = i/29$ for $1 \leq i \leq 29$.

When using the GDDS algorithm to solve random *Watson function* minimization problem $\min_{z \in \mathbb{R}^n} \mathbb{E}[f(z,\xi)]$, the initial point is set at $(0.5, \cdots, 0.5)^\top$ with an appropriate dimension. Let $\Delta_0 = 1$, $\Delta_{tol} = 0.001$, $N_0 = 5n$, $\mathcal{D}_k = \{\pm e_i | i = 1, \cdots, n\}$, and the other parameters are same with the VNSP3 in Section 5.1. We report the step-length Δ_k, the estimated function values and the sample size N_k in each iteration k in Figs. 3-4.

From the results, we can see that the growth of the problem dimension (from 10 to 30) leads to the increment of the number of iterations before satisfying the same stop tolerance. Note also that in the early stages the GDDS algorithm reduces function values in a fast pace while the sample size remain unchanged, which means that the sample sizes in the VNSP3 is increased conservatively at the initial stages despite of its overall efficiency.

5.3. Comparative analysis

Before the comparative analysis, we need to emphasize again that StoMADS [1] is proposed within a blackbox framework without assuming the ability of fixing the sample, which shows that StoMADS is feasible for more problems compared to GDDS algorithm. In this subsection, we focus on the efficiency of each algorithm.

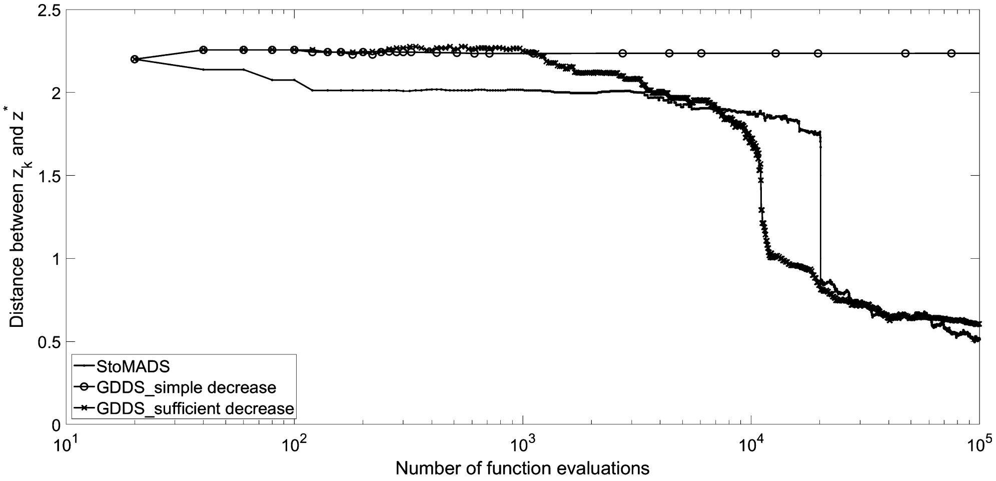

Fig. 5. Comparison on the distance of z_k in GDDS and StoMADS to the true solution: additive noise.

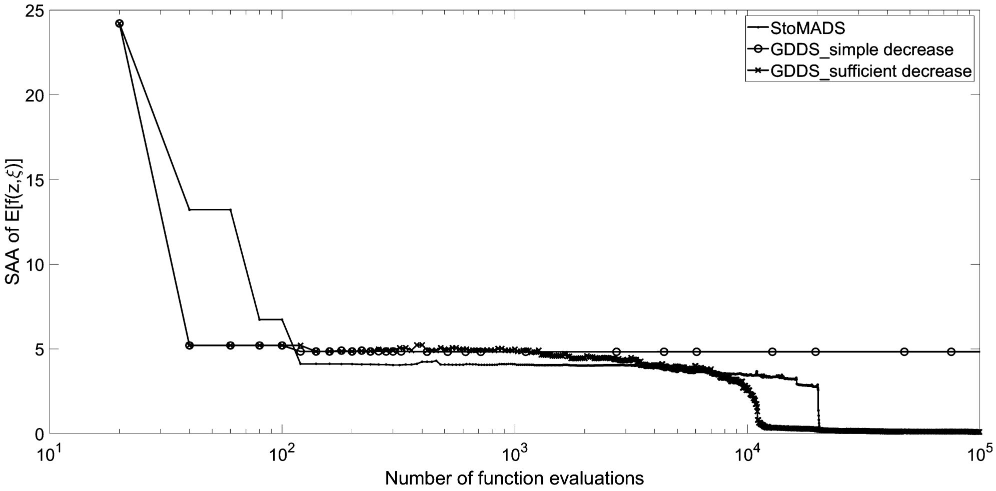

Fig. 6. Comparison on the function value at z_k in GDDS and StoMADS: additive noise.

StoMADS is an iterative algorithm designed for unconstrained optimization of function with random noise. In each iteration, StoMADS has two main steps: search step and poll step. Since the search step in StoMADS is optional and we focus on the comparison of the GDDS and the StoMADS with the same parameter settings, the search steps are not included in the StoMADS when solving these numerical tests. The main difference between GDDS and StoMADS is the sampling scheme. The GDDS updates sample and increases its size at each unsuccessful iterations. Hence the sample sizes are consecutively increased when approaching the stationary points.

To make two different approaches comparable, we focus our analysis on the scenarios where different samples are used for evaluating different trial solutions at the same iteration. To this end, we modify Algorithm 1 by generating a new sample for each trial solutions but control the sample sizes being the same in an iteration. This modification might impact the validity of our convergence analysis on the GDDS in Section 4, where additional samples in each iteration might introduce higher variances and hence less effectiveness in selecting the right candidate. Within this framework, we implement StoMADS and GDDS to minimize the random Rosenbrock function with “additive noise”, in which

$$f(z,\xi) = \left(10\left(z_2 - z_1^2\right) + \zeta_1\right)^2 + \left((1-z_1)^2 + \zeta_2^2\right)^2$$

where ζ_1, ζ_2 are independent random variables with a uniform distribution from $[-0.100, 0.100]$. Since the true problem is $\min_{z\in\mathbb{R}^n} \mathbb{E}[f(z,\zeta)]$, we set stop criterion as

$$\frac{f(z_k) - f(z^*)}{f(z_0) - f(z^*)} \leq \tau, \tag{11}$$

where z_k is the incumbent solution in iterate k, $z_0 = (-1.200, 1.000)^\top$, $z^* = (1.000, 1.000)^\top$ and $\tau = 0.010$.

The results for the “additive noise” scenarios are presented in the following two figures: Fig. 5 illustrates that GDDS with simple decrease stops at a solution with sufficiently large distance from the true solution where the true solution of this problem analytically obtainable. On the contrary, StoMADS exhibits a faster convergence pace at the beginning iterations, and finally StoMADS and GDDS with sufficient decrease reach almost the same convergence rate. Similar results can be also observed in Fig. 6.

Now we proceed to more complex problem of minimizing the random Rosenbrock function with “multiplicative noise” in Section 5.1. It is worth to mention that StoMADS is not intentionally proposed for solving this type of problems. We perform these comparative analysis to illustrate differences between StoMADS and different types of GDDS. First, we consider the scenarios where the sample can be fixed and used for every trial solutions where we modify the original StoMADS by repeatedly implementing the

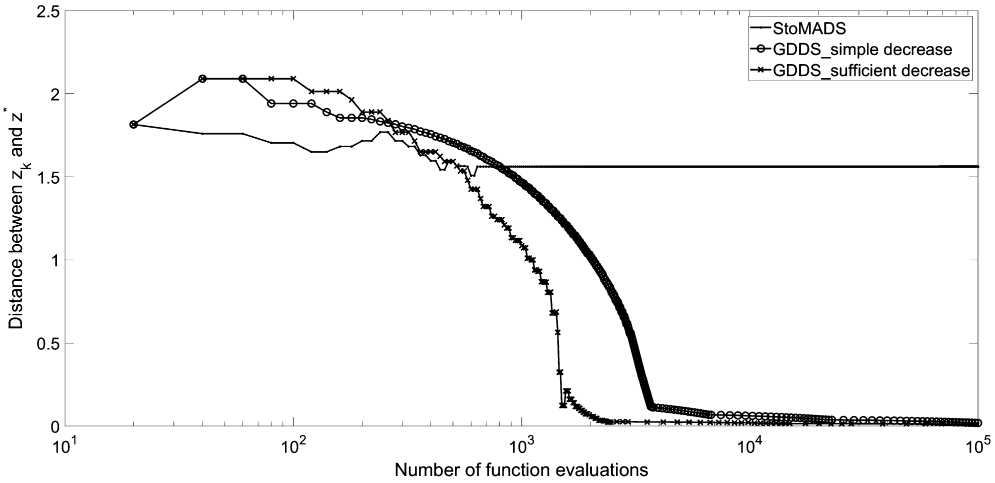

Fig. 7. Comparison on the distance of z_k in GDDS and StoMADS to the true solution: multiplicative noise and fixed sample.

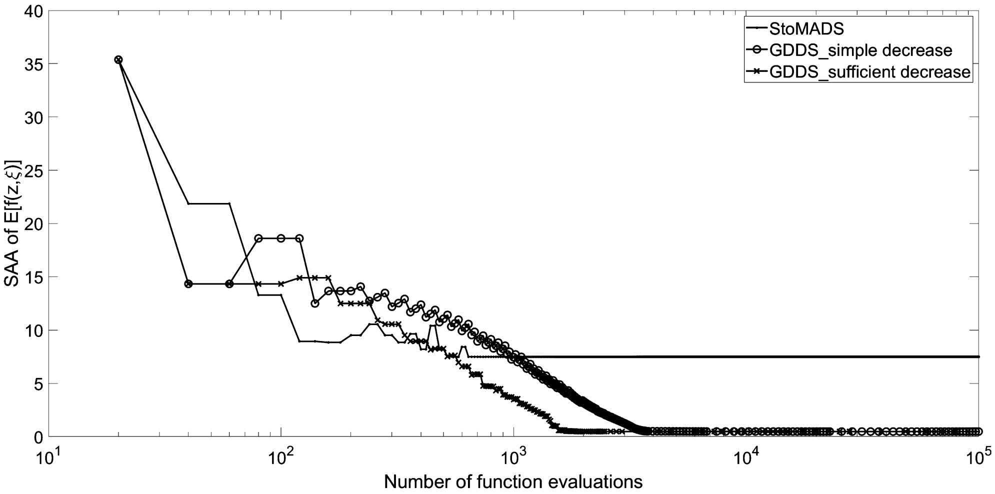

Fig. 8. Comparison on the function value at z_k in GDDS and StoMADS: multiplicative noise and fixed sample.

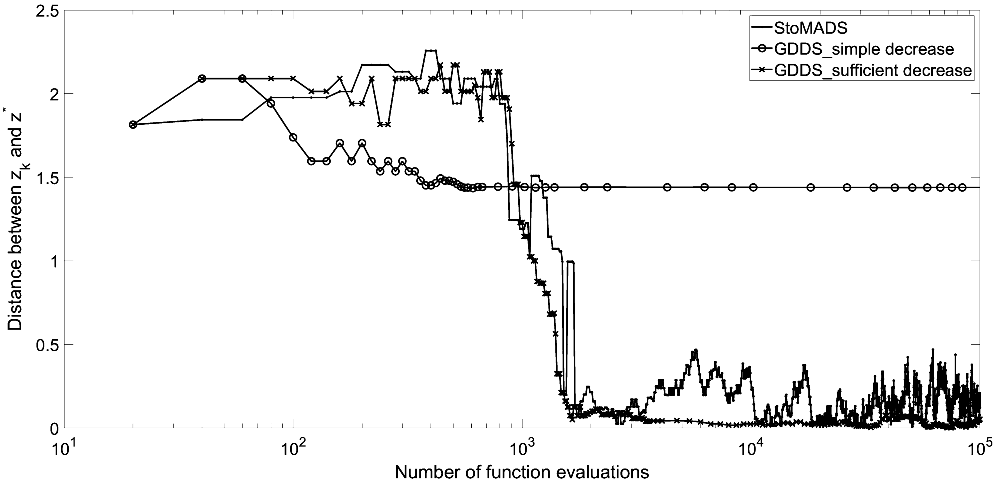

Fig. 9. Comparison on the distance of z_k in GDDS and StoMADS to the true solution: multiplicative noise and different sample.

same sample within an iteration. The following figures exhibits the performances of different methods in the distances of z_k to the true solution z^* and the function values at each iterate. From Figs. 7 and 8 we can observe that the StoMADS stops at solutions with certain distances from the true solution but two GDDS methods converges. In addition, GDDS with sufficient decreases converges with less number of observations being used. The phenomena in these figures might implies that the scheme of fixing sample in every iteration not necessarily satisfies the convergence conditions in StoMADS.

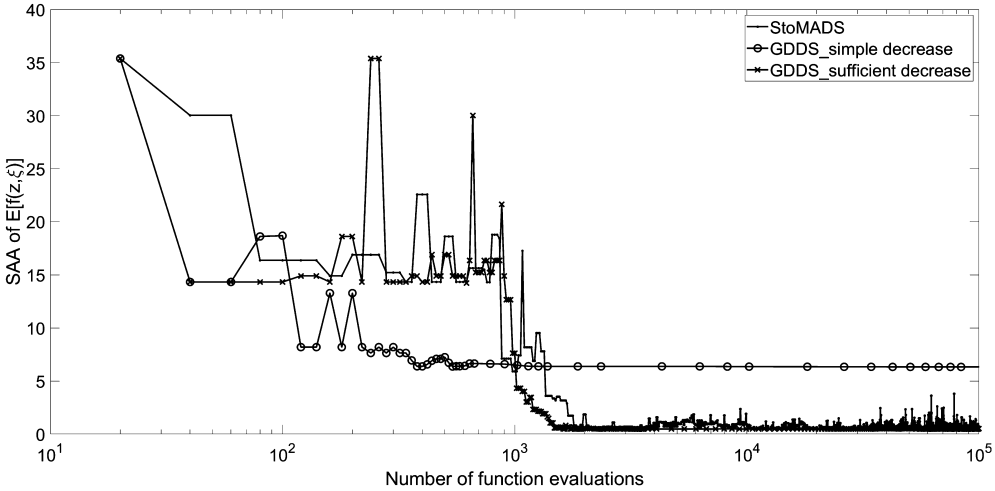

Fig. 10. Comparison on the function value at z_k in GDDS and StoMADS: multiplicative noise and different sample.

We further consider a new scenario in which a new sample has to be generated and implemented to evaluate a new trial solution even when the iteration is unchanged. We again compare the distance of z_k attained from three methods to the true optimal solutions and the function values of z_k. Figs. 9 and 10 clearly show that StoMADS exhibits a much better performance compared to its counterpart in the fixed sample scheme in Fig. 7 and 8. It can be recognized that the reduce rate of function values in StoMADS is at the same level as the GDDS with sufficient decrease, while the GDDS with simple decrease fails in the convergence. At the stage when the number of observations used more than 10^3, z_k yielded in each iteration of StoMADS is more volatile than its counterparts in GDDS with sufficient decrease, but the higher volatility in z_k at this stage does not affect the function value where the standard deviations of z_k yielded by StoMADS is slightly higher.

6. Conclusion

The paper has proposed a sampling scheme to broaden the well-known DDS algorithm to solve stochastic optimization problems where the objective functions are not analytical obtainable. To minimize the sampling and computation load, the framework of the variable sampling scheme has been incorporated into the DDS algorithm and the conditions on the algorithm convergence have been investigated. The condition on the growth of sample in each iteration has been given to guarantee the convergence of the DDS to the true optimal solution with probability one. One of the open questions is how to determine which method is of best effectiveness in saving the sampling efforts and how to allocate the samples in each iteration of the method under a given sample resource.

Acknowledgement

The authors are grateful to the anonymous referee and the handling Editor for their helpful comments and suggestions. The authors appreciate the support from Shenzhen Fundamental Research Program (JCYJ 20190808164605481), National Natural Science Foundation of China (No. 72071128, 71931007), and National Natural Science Foundation of China with the Research Project on Human-Machine Collaborative Decision-Making (No. 72192822).

Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.orl.2022.03.006.

References

[1] C. Audet, K.J. Dzahini, M. Kokkolaras, S. Le Digabel, Stochastic mesh adaptive direct search for blackbox optimization using probabilistic estimates, Comput. Optim. Appl. 79 (2021) 1–34.

[2] F. Bastin, C. Cirillo, P.L. Toint, An adaptive Monte Carlo algorithm for computing mixed logit estimators, Comput. Manag. Sci. 3 (1) (2006) 55–79.

[3] G. Bayraksan, D.P. Morton, A sequential sampling procedure for stochastic programming, Oper. Res. 59 (4) (2011) 898–913, https://doi.org/10.1287/opre.1110.0926.

[4] H. Chen, B.W. Schmeiser, Stochastic root finding via retrospective approximation, AIIE Trans. 33 (3) (2001) 259–275, https://doi.org/10.1023/A:1007663818987.

[5] R. Chen, M. Menickelly, K. Scheinberg, Stochastic optimization using a trustregion method and random models, Math. Program. 169 (2) (2018) 447–487.

[6] X. Chen, C.T. Kelley, Optimization with hidden constraints and embedded Monte Carlo computations, Optim. Eng. 17 (1) (2016) 157–175, https://doi.org/10.1007/s11081-015-9302-1.

[7] G. Deng, M.C. Ferris, Variable-number sample-path optimization, Math. Program. 117 (1–2) (2009) 81–109, https://doi.org/10.1007/s10107-007-0164-y.

[8] N.I.M. Gould, D. Orban, P.L. Toint, Cutest: a constrained and unconstrained testing environment with safe threads for mathematical optimization, Comput. Optim. Appl. 60 (3) (2015) 545–557, https://doi.org/10.1007/s10589-014-9687-3.

[9] T. Homem-de Mello, Variable-sample methods for stochastic optimization, ACM Trans. Model. Comput. Simul. 13 (2) (2003) 108–133, https://doi.org/10.1145/858481.858483.

[10] S. Kim, D. Zhang, Convergence Properties of Direct Search Methods for Stochastic Optimization, 2010, pp. 1003–1011, https://doi.org/10.1109/WSC.2010.5679089.

[11] T.G. Kolda, R.M. Lewis, V. Torczon, Optimization by direct search: new perspectives on some classical and modern methods, SIAM Rev. 45 (3) (2003) 385–482, https://doi.org/10.1137/S003614450242889.

[12] J. Larson, M. Menickelly, S.M. Wild, Derivative-free optimization methods, Acta Numer. 28 (2019) 287–404, https://doi.org/10.1017/S0962492919000060.

[13] R.M. Lewis, V. Torczon, M.W. Trosset, Direct search methods: then and now, J. Comput. Appl. Math. 124 (1–2) (2000) 191–207, https://doi.org/10.1016/S0377-0427(00)00423-4.

[14] R. Pasupathy, On choosing parameters in retrospective-approximation algorithms for stochastic root finding and simulation optimization, Oper. Res. 58 (2010) 889, https://doi.org/10.1287/opre.1090.0773.

[15] R. Pasupathy, B.W. Schmeiser, Retrospective-approximation algorithms for the multidimensional stochastic root-finding problem, ACM Trans. Model. Comput. Simul. 19 (2) (2009), https://doi.org/10.1145/1502787.1502788.

[16] J.O. Royset, On sample size control in sample average approximations for solving smooth stochastic programs, Comput. Optim. Appl. 55 (2) (2013) 265–309, https://doi.org/10.1007/s10589-012-9528-1, cited by 9.

[17] A. Shapiro, T. Homem-de Mello, A simulation-based approach to two-stage stochastic programming with recourse, Math. Program., Ser. B 81 (3) (1998) 301–325, https://doi.org/10.1007/BF01580086.

[18] A. Shapiro, D. Dentcheva, A. Ruszczynski, Lectures on Stochastic Programming, second edition, SIAM, Philadelphia, PA, USA, 2014.