Journal of Global Optimization manuscript No.
(will be inserted by the editor)

Filter-based stochastic algorithm for global optimization

M. Joseane F. G. Macêdo · Elizabeth W. Karas · M. Fernanda P. Costa · Ana Maria A. C. Rocha

Received: date / Accepted: date

Abstract We propose the general Filter-based Stochastic Algorithm (FbSA) for the global optimization of nonconvex and nonsmooth constrained problems. Under certain conditions on the probability distributions that generate the sample points, almost sure convergence is proved. In order to optimize problems with computationally expensive black-box objective functions, we develop the FbSA-RBF algorithm based on the general FbSA and assisted by Radial Basis Function (RBF) surrogate models to approximate the objective function. At each iteration, the resulting algorithm constructs/updates a surrogate model of the objective function and generates trial points using a dynamic coordinate search strategy similar to the one used in the Dynamically Dimensioned Search method. To identify a promising best trial point, a non-dominance concept based on the values of the surrogate model and the constraint violation at the trial points is used. Theoretical results concerning the sufficient conditions for the almost surely convergence of the algorithm are presented. Preliminary numerical experiments show that the FbSA-RBF is competitive when compared with other known methods in the literature.

Keywords Global optimization · Dynamically dimensioned search · Filter method.

M. Joseane F. G. Macêdo
Dep. of Exact and Natural Sciences, Federal Rural University of Semi-Árido, Mossoró, Brazil
E-mail: joseane@ufersa.edu.br

Elizabeth W. Karas
Dep. of Mathematics, Federal University of Paraná, Curitiba, Brazil
E-mail: ewkaras@ufpr.br

M. Fernanda P. Costa
Centre of Mathematics, University of Minho, Campus de Gualtar, Braga, Portugal
E-mail: mfc@math.uminho.pt

Ana Maria A. C. Rocha
ALGORITMI Center, University of Minho, Campus de Gualtar, Braga, Portugal
E-mail: arocha@dps.uminho.pt

1 Introduction

This paper aims to contribute to the research area of global optimization by proposing a general Filter-based Stochastic Algorithm (FbSA) for nonconvex and nonsmooth constrained optimization problems. One of the main goals of this paper is to prove that, under certain hypotheses on the probability distributions that generate the sample points, the algorithm converges to a global optimum of the problem in a probabilistic sense. Since many real-world optimization problems involve black-box objective functions whose objective function values are obtained through computationally expensive simulations, another main goal of this paper is to develop a FbSA assisted by Radial Basis Function (RBF) surrogate models to approximate the objective function to conform to the proposed general FbSA. We denote this algorithm as FbSA-RBF. The mathematical formulation of the problem is

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & c_i(x) \leq 0,\ i \in \mathcal{I} \\ & c_i(x) = 0,\ i \in \mathcal{E} \\ & x \in \mathcal{D} \end{array} \tag{1}$$

where $f : A \subseteq \mathbb{R}^n \to \mathbb{R}$ is a deterministic function defined on a set $A \supseteq \mathcal{D}$ of the $\mathbb{R}^n$, $c_i : \mathbb{R}^n \to \mathbb{R}$, $i \in \mathcal{I} \cup \mathcal{E}$ are the constraint functions, and $\mathcal{D} \subset \mathbb{R}^n$ is a finite box of the form $\mathcal{D} = \{x \in \mathbb{R}^n : l_i \leq x_i \leq u_i, \forall i = 1, 2, \ldots, n\}$, and all lower and upper bounds l_i and u_i are finite for $i = 1, \ldots, n$. The feasible region is denoted by $\mathcal{D}_V := \{x \in \mathcal{D} : c_i(x) \leq 0,\ i \in \mathcal{I},\ c_i(x) = 0,\ i \in \mathcal{E}\}$. We assume that f is bounded below, and there exists a global minimizer x^* of f over $\mathcal{D}_V$ with f continuous at point x^*. It is also assumed that f and c_i, with $i \in \mathcal{I} \cup \mathcal{E}$, are Lebesgue measurable.

Many different approaches have been proposed for the global optimization of (1). Among these are the penalty methods, which reformulate (1) to the minimization of an appropriate penalty function over $\mathcal{D}$ [3,5,9,12,23,32]. Mathematically, the penalty function is defined by the objective function and the constraints violation (penalty terms) multiplied by some positive penalty parameter. However, initializing the penalty parameter and adjusting its value throughout the iterative process are problematic issues since their values affect the performance of the algorithms. In some cases, the optimal solution is obtained when the penalty parameter tends towards infinity.

There are in the literature other penalty functions, namely the exact penalty functions and the augmented Lagrangian functions for which, in general, it is sufficient a finite parameter value to guarantee convergence to the optimal solution of problem (1).

Exact penalty functions and augmented Lagrangian functions have been extended for global optimization (see [6,12,19,47], and references therein). For example, in [12], an Exact Penalty Global Optimization (EPGO) algorithm is proposed, where an improved version of the deterministic DIRECT algorithm [18] is used to globally solve a finite sequence of bound constrained subproblems. The bound constrained subproblems are defined by a non-differentiable exact penalty function for given values of the penalty parameter and an automatic updating rule of the penalty parameter that occurs only a finite number of times. It is proved that EPGO produces a sequence of optimal solutions of the underlying subprob-

lems such that any limit point of the sequence is a global solution of the related constrained problem.

Other strategies for handling the constraints of the problem, that avoid updating the penalty parameters associated to the penalty functions, have been proposed in the literature. For instance, techniques based on multi-objective concepts in which the constrained problem is reformulated as a bi/multi-objective one, being the objective function and the infeasibility measures minimized simultaneously [1, 2,17]. One of the most known is the filter method introduced by Fletcher and Leyffer in [17], which guarantee sufficient progress towards feasibility and optimality of nonlinear constrained problems. The filter method uses the concept of dominance, borrowed from multi-objective optimization, to build a filter that accepts iterates if they improve the objective function or improve a constraint violation function based on the Pareto dominance rule. Ever since, an abundance of filter-based approaches have been proposed [4,10,11,14,15,21,31,46,48].

In particular, the filter methods have been also incorporated into stochastic algorithms [24,28,34,42], and the resulting algorithms have been shown that the filter methodology is promising when compared to other strategies. For example, in [42], a filter methodology is incorporated into the population-based artificial fish swarm (P-BF AFS) algorithm for globally solving nonsmooth constrained optimization problems. The constrained problem is replaced by a sequence of bound constrained bi-objective subproblems. The P-BF AFS is used to solve the bound constrained bi-objective subproblems with an increasing accuracy required. The filter methodology is used to accept non-dominated trial points that improve over the current ones.

In [28] is presented a method that combines the filter method and the Dynamically Dimensioned Search (DDS) method [45], for solving nonsmooth constrained global optimization problems. The proposed algorithm reformulates the constrained optimization problem as a bound constrained bi-objective one and uses the filter method to control the progress related to optimality and feasibility by defining a forbidden region of points using the flat or slanting filter rule [28]. At each iteration, the algorithm computes a set of trial points using a similar strategy to DDS to explore the search space for a global optimum. The filter methodology is used to accept the best non-dominated trial point that improves related to the best current one. The resulting algorithm is refereed as A2-FF or A2-SF if it is used the flat or slanting filter rule, respectively. The A2-FF/SF is enriched with a poll-strategy which is invoked whenever a best non-dominated point is not found. This procedure searches in a vicinity of the least infeasible non-dominated point found so far, with the hope to approach a different part of the feasible region of the problem.

It is noteworthy that the DDS is a stochastic global algorithm [45] developed for solving calibration problems that arise in the context of watershed simulation models. These problems are bound constrained ones, with many parameters/variables and computationally expensive objective functions. At each iteration, the DDS generates a randomly trial point by perturbing some coordinates of the current best point, which are dynamically and probabilistically chosen. At the beginning, the DDS searches globally, and becomes a more local search as the number of functions evaluations approaches the maximum allowed. The transition from global to local search is achieved by dynamically and probabilistically reducing the number of dimensions to be perturbed in the neighborhood of the current best solution.

Since DDS has shown to be a simple and robust tool for solving computationally expensive calibrations problems, a variety of new approaches or extensions that use a dynamic coordinate search strategy similar to the one used in DDS have been proposed in the literature [36,37,38,39,44]. For example, in [44], the DDS algorithm was modified to solve discrete, single-objective, constrained water distribution system design problems. The ConstrLMSRBF algorithm for large-scale optimization involving expensive black-box objective and constraint functions is proposed in [36]. This algorithm is an extension of the Local Metric Stochastic RBF (LMSRBF) algorithm [38] designed for bound constrained optimization problems with an expensive black-box objective function. The ConstrLMSRBF uses RBF surrogate models for the objective function and also for all the constraint functions. The ConstrLMSRBF requires a feasible starting point and it always attempts to maintain feasibility along the entire iterative process.

In this paper, we aim to propose a general FbSA for solving nonconvex and nonsmooth constrained global optimization problems. This study comes in sequence of our previous preliminary practical study concerning the filter and DDS methods [28]. Under certain conditions on the probability distributions that generate the points, almost sure convergence is proved for a class of algorithms conforming to this general algorithm. In order to solve problems with computationally expensive black-box objective functions, a FbSA assisted by Radial Basis Function surrogate models to approximate the objective function, according to the general FbSA, is presented.

The paper is organized as follows. In Section 2, the general FbSA is presented and its almost sure convergence is established. Section 3 describes the FbSA-RBF assisted by surrogate models to approximate the objective function, and presents its convergence analysis. Section 4 shows the numerical results and the paper is concluded in Section 5.

2 General filter-based stochastic algorithm

In this section, we present the general FbSA for solving nonconvex and nonsmooth constrained global optimization problem (1). The framework of the embedded filter algorithm is similar to [25, Alg. 2.1]. Due to stochastic nature of the algorithm, the iterates are treated as random vectors whose realizations are in $\mathcal{D}$, following the ideas of [35]. The algorithm is general since the iterates are given randomly by any probability distribution. The global convergence of the FbSA in the probabilistic sense is established.

2.1 The algorithm

The general FbSA generates two sequences of random vectors whose realizations are in $\mathcal{D} \subset \mathbb{R}^n$. Namely, the sequence of iterates (X_k) and the sequence of the best iterates (X_k^*) selected from (X_k).

The sequence of iterates (X_k) is defined on a probability space $(\Omega, \mathcal{B}, P)$, where Ω is the sample space, $\mathcal{B}$ is a σ-algebra and $P : \mathcal{B} \to [0, 1]$ is a probability function. The random vector $X_k : (\Omega, \mathcal{B}) \to (\mathcal{D}, \mathcal{B}(\mathcal{D}))$ represents the kth function evaluation

iterate, where $\mathcal{B}(\mathcal{D})$ are the Borel sets in $\mathcal{D}$. At iteration k, to define X_k, a collection of random elements is used, herein denoted by

$$\{\Lambda_{k_j} \,:\, (\Omega, \mathcal{B}) \to (\Omega_{k_j}, \mathcal{B}_{k_j}), \text{ for } j = 1, \ldots, \ell_k\}$$

where $\ell_k > 1$ is the number of random elements that are generated. The FbSA is general in the sense that to define the random vectors X_k there exists a certain freedom to choose the probability space $(\Omega, \mathcal{B}, P)$ of the collection of random elements.

To simplify the description of general FbSA, the following notation is used. Given an initial random vector $X_0 \in \mathcal{D}$, the *oracle* associated to the collection of random elements is defined as follows: $\mathcal{O}_0 = \{X_0\}$ and for each iteration $k \geq 1$

$$\mathcal{O}_k = \{\Lambda_{t_j} \,:\, t = 1, \ldots, k \text{ and } j = 1, \ldots, \ell_k\} = \mathcal{O}_{k-1} \cup \{\Lambda_{k_1}, \cdots, \Lambda_{k_{\ell_k}}\}. \quad (2)$$

Hence, in the general FbSA, using a deterministic function $\Theta : \mathcal{O}_k \to \mathcal{D}$, the random vector iterate is computed as $X_k = \Theta(\mathcal{O}_k)$ from the random elements of $\mathcal{O}_k$. We note that, in general, a realization of X_k can fall outside of the domain $\mathcal{D}$. To prevent this, the deterministic function has the property to transform any point of $\mathbb{R}^n$ into a suitable point of $\mathcal{D}$. Since $\mathcal{D}$ is compact, the function $\Theta(\mathcal{O}_k)$ can involve a projection into $\mathcal{D}$ whenever $X_k \notin \mathcal{D}$.

The progress of the random vector iterates X_k, $k \geq 1$, is controlled by a filter methodology as suggested in [8,25]. The filter methodology treats the optimization problem (1) as bi-objective, and attempts to minimize both the objective function f and a nonnegative aggregate constraint violation function h.

Herein, the constraint violation function $h : \mathbb{R}^n \to \mathbb{R}^+$ is defined by

$$h(x) = \|c^+(x)\|, \quad (3)$$

where $\|\cdot\|$ is an arbitrary norm and $c^+ : \mathbb{R}^n \to \mathbb{R}^m$ is given by

$$c_i^+(x) = \begin{cases} c_i(x), & \text{if } i \in \mathcal{E} \\ max\{0, c_i(x)\}, & \text{if } i \in \mathcal{I}. \end{cases}$$

The filter method is an efficient technique that builds the forbidden region of the points, known as dominated region. The filter is defined as a set of pairs $\{(f_j, h_j)\}$ from the former iterations.

The general FbSA deals with the filter and with the forbidden region associated with it, as follows. At iteration $k = 0$, given an initial random vector $X_0 \in \mathcal{D}$, we set $X_0^* = X_0$, and we initialize the filter $F_k = \emptyset$ and define the forbidden region $\mathcal{F}_k = \emptyset$.

At the beginning of each iteration $k \geq 1$, the pair $(f(X_{k-1}^*), h(X_{k-1}^*))$ from the previous best iterate is temporarily added to the current filter, $\bar{F}_{k-1} \equiv F_{k-1} \cup \{(f(X_{k-1}^*), h(X_{k-1}^*))\}$. As proposed in [8], to avoid convergence to infeasible accumulation points X^* where $h(X^*) > 0$, we add an envelope around the temporary current filter $\bar{F}_{k-1}$, and a pair $(f(x), h(x))$ is non-dominated by any pair of the filter, if for all $(f_j, h_j) \in \bar{F}_{k-1}$

$$f(x) + \alpha h(x) \leq f_j \quad \text{or} \quad h(x) \leq (1 - \alpha) h_j \quad (4)$$

where $\alpha \in (0,1)$ is a given constant. A filter based on this acceptance rule is called by *slanting filter*. Taking into account the acceptance rule (4), the forbidden region is defined by

$$\bar{\mathcal{F}}_{k-1} = \mathcal{F}_{k-1} \cup \left\{ x \in \mathcal{D} : \; f(x) + \alpha h(x) \geq f(X_{k-1}^*) \;\text{ and }\; h(x) \geq (1-\alpha) h(X_{k-1}^*) \right\}.$$

Thus, a realization of X_k is computed such that the pair $(f(X_k), h(X_k))$ is acceptable by the filter $\bar{F}_{k-1}$, or equivalently $X_k \notin \bar{\mathcal{F}}_{k-1}$. One must keep in mind that the forbidden region is never constructed in the algorithm, but helps the understanding of the process.

When X_k produces a reduction on f, the iteration is called an f-type iteration. Otherwise, it is called an h-type iteration [20,31]. This classification is used for updating the filter. Herein, the filter is updated only at h-type iterations, in such iterations the pair $(f(X_{k-1}^*), h(X_{k-1}^*))$ becomes permanent in the filter F_k, and all pairs that are dominated by $(f(X_{k-1}^*), h(X_{k-1}^*))$ can be removed from the filter. If the filter is not updated, it remains unchanged, and the pair $(f(X_{k-1}^*), h(X_{k-1}^*))$ is discarded. Thus, the filter is updated as follows:

$$F_k = \begin{cases} \bar{F}_{k-1}, \text{ if } k \text{ is an } h\text{-type iteration} \\ F_{k-1}, \text{ otherwise.} \end{cases}$$

This updating rule prevents the addition of feasible points to the filter. Note that, if X_{k-1}^* is feasible then X_k must decrease f to be accepted by the filter $F_{k-1} \cup \{(f(X_{k-1}^*), 0)\}$. Therefore, the iteration must be an f-type iteration, and consequently the pair $(f(X_{k-1}^*), 0)$ is discarded. Furthermore, this updated rule assures movements towards some infeasible regions to return back later to a different part of the feasible one.

The new iterate $X_k \in \mathcal{D}$ accepted by the filter will be selected as a new best point if it does not increase the infeasibility measure more than a small positive quantity $\varepsilon_h > 0$, from the current best point. Figure 1 illustrates the region of pairs (f, h) that are forbidden or dominated by $(f(X_{k-1}^*), h(X_{k-1}^*))$ and the region of pairs that are considered best than it. Allowing a small controlled increase in the infeasibility measure h with a reduction in the optimality measure f, this strategy aims to promote the exploration of other potentially promising neighborhood areas of the feasible region and to prevent convergence to optimal local solutions. Note that once a best feasible point is obtained, the best point will be updated only if the new one reduces the objective function value.

A formal description of the general FbSA for solving problem (1) is presented in Algorithm 1.

2.2 Global convergence

In this section we present the convergence analysis of Algorithm 1 in the probabilistic sense.

The first lemma ensures that all elements of the filter correspond to infeasible points and that the algorithm is well-defined in the sense that whenever the current point does not satisfy the stop criterion, a new not forbidden point can be computed.

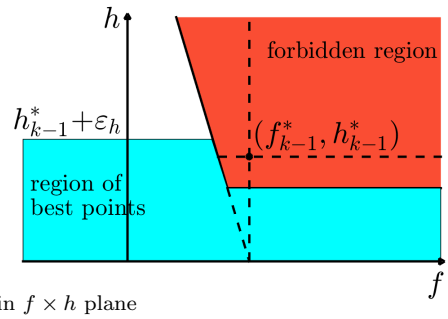

Fig. 1 Filter criterion in $f \times h$ plane

Algorithm 1 - FbSA

Require: $\alpha \in (0,1)$, $\varepsilon_h > 0$

randomly generate a point $X_0 \in \mathcal{D}$
set $X_0^* = X_0$, $\mathcal{O}_0 = \{X_0^*\}$ and $F_0 = \mathcal{F}_0 = \emptyset$
set $k = 1$
while the stopping criterion is not satisfied **do**
  set $\bar{F}_{k-1} = F_{k-1} \cup \{(f(X^*_{k-1}), h(X^*_{k-1}))\}$ and define
  $\bar{\mathcal{F}}_{k-1} = \mathcal{F}_{k-1} \cup \left\{x \in \mathcal{D} : f(x) + \alpha h(x) \geq f(X^*_{k-1}) \text{ and } h(x) \geq (1-\alpha)h(X^*_{k-1})\right\}$
  generate a collection of random elements $\{\Lambda_{k_j}\}_{1 \leq j \leq \ell_k}$ according to some probability distribution
  set $\mathcal{O}_k = \mathcal{O}_{k-1} \cup \{\Lambda_{k_j}\}_{1 \leq j \leq \ell_k}$
  compute $X_k = \Theta(\mathcal{O}_k)$ such that $X_k \in \mathcal{D}$ and $X_k \notin \bar{\mathcal{F}}_{k-1}$
  Filter update:
  if $f(X_k) \leq f(X^*_{k-1})$ **then**
    set $F_k = F_{k-1}$ and $\mathcal{F}_k = \mathcal{F}_{k-1}$ (f-iteration)
  else
    set $F_k = \bar{F}_{k-1}$ and $\mathcal{F}_k = \bar{\mathcal{F}}_{k-1}$ (h-iteration)
  end if
  Best point update:
  if $h(X_k) < h(X^*_{k-1}) + \varepsilon_h$ **then**
    set $X^*_k = X_k$ (successful iteration)
  else
    set $X^*_k = X^*_{k-1}$ (unsuccessful iteration)
  end if
  $k = k + 1$
end while

Lemma 1 *For all $k \geq 1$ such that X_{k-1} does not satisfy the stopping criterion of Algorithm 1, the following two statements hold:*

(i) $h(X_j) > 0$, *for all* $(f(X_j), h(X_j)) \in F_{k-1}$;
(ii) There exists $X_k \notin \bar{\mathcal{F}}_{k-1}$.

Proof Due to the general FbSA framework and the embedded filter methodology that is similar to the one presented in [25], the proof follows analogous to [25, Alg. 2.1]. □

The next theorem states that any accumulation point of the sequences generated by the algorithm is feasible.

Theorem 1 *Consider the sequences (X_k) and (X_k^*) generated by Algorithm 1. Then $h(X_k) \to 0$ and $h(X_k^*) \to 0$. In addition, the number of unsuccessful iterations is finite.*

Proof The result for the sequence $h(X_k)$ follows from [25, Thm. 2.3]. By the selection of the best point in the algorithm,

$$h(X_k^*) = h(X_k) \quad \text{or} \quad h(X_k^*) = h(X_{k-1}^*) \leq h(X_k) - \varepsilon_h < h(X_k),$$

where $\varepsilon_h > 0$. So, $0 \leq h(X_k^*) \leq h(X_k)$, for all $k \in \mathbb{N}$, and consequently $h(X_k^*) \to 0$.

Now suppose by contradiction that the number of unsuccessful iterations is infinite. In this case, there exists a subsequence of indices $(k_j)_{j\geq 1}$ satisfying,

$$h(X_{k_j}) \geq h(X_{k_j-1}^*) + \varepsilon_h \quad \text{and} \quad h(X_{k_j}^*) = h(X_{k_j-1}^*).$$

Thus, $h(X_{k_j}) \geq h(X_{k_j}^*) + \varepsilon_h$. As $h(X_k)$ and $h(X_k^*)$ converge to zero, setting $j \to \infty$, it follows that $\varepsilon_h \leq 0$, which contradicts the fact that ε_h is a strictly positive constant. □

Now we analyse the convergence of Algorithm 1 in the probabilistic sense following the ideas of [35]. We say that an algorithm converges to the global minimum of f on $\mathcal{D}$ in probability, or almost surely, if the sequence $(f(X_k^*))$ converges to f^* in probability, or almost surely.

Let $\sigma(\mathcal{O}_k)$ be the σ-algebra generated by the oracle which represents all the information that can be derived from the random elements in $\mathcal{O}_k$.

In the light of Lemma 1, we shall suppose that Algorithm 1 generates infinite sequences (X_k) and (X_k^*). Furthermore, let μ be the Lebesgue measure in $\mathbb{R}^n$ and assume the following hypotheses.

H1 *The function f has a global minimizer x^* over $\mathcal{D}_V$, with f continuous at point x^*. The functions f and c_i, $i \in \mathcal{I} \cup \mathcal{E}$, are Lebesgue measurable.*

H2 *For all $\rho, \varepsilon > 0$, $\mu(T_{\rho,\varepsilon}) > 0$, where*

$$T_{\rho,\varepsilon} := \{x \in \mathcal{D} : h(x) \leq \rho \text{ and } f(x) < f^* + \varepsilon\}. \tag{5}$$

H3 *For all $\delta > 0$,*

$$\psi_{\mathcal{D}}(\delta) := \inf_{z\in\mathcal{D}} \mu\left(B(z,\delta) \cap \mathcal{D}\right) > 0.$$

The Hypotheses H1 and H2 are satisfied as long as $\mathcal{D}_V$ is a bounded region and the functions f and c_i, $i \in \mathcal{I}\cup\mathcal{E}$, are continuous. Hypothesis H2 is a key assumption and quite usual in this context [34,42]. It precludes f from having an isolated discontinuous optimum point (for more details, see [22]). In [34], the convergence is guaranteed almost surely for a point in the set $S^* = \{x \in \mathcal{D}_V : f(x) \leq f_{egs}^*\}$, where f_{egs}^* is the infimum on all values f_t for which

$$\mu\left(\{z \in \mathcal{D} : h(z) \leq \rho,\, f(z) < f_t\}\right) > 0,$$

for all $\rho > 0$. Hypothesis H3 is also considered in [35] and guarantees that the intersection of $\mathcal{D}$ and any ball centered in a point of $\mathcal{D}$ has positive measure. This condition is important to prove the next lemma.

Lemma 2 *Consider the sequence (X_k) generated by Algorithm 1. Let $(X_{k_j})_{j\geq 1}$ be a subsequence of (X_k) such that for each $j \geq 1$, X_{k_j} has a conditional density*

$$g_{k_j}(x \,|\, \sigma(\mathcal{O}_{(k_j)-1})).$$

Suppose that

$$\mu(\{x \in \mathcal{D} \,:\, G(x) = 0\}) = 0, \quad \textit{where} \quad G(x) := \inf_{j\geq 1} g_{k_j}(x \,:\, \sigma(\mathcal{O}_{(k_j)-1})). \tag{6}$$

Then, the following statements hold:

(i) For all $z \in \mathcal{D}$ and $\delta > 0$, there exists $v(z,\delta) \in (0,1)$ such that

$$P\left(X_{k_j} \in B(z,\delta) \cap \mathcal{D} \,|\, \sigma(\mathcal{O}_{(k_j)-1})\right) \geq v(z,\delta). \tag{7}$$

(ii) For all $\varepsilon > 0$, there exists $L(\varepsilon) \in (0,1)$ such that

$$P\left(X_{k_j} \in \mathcal{D} :\, f(X_{k_j}) < f(x^*) + \varepsilon \,|\, \sigma(\mathcal{O}_{(k_j)-1})\right) \geq L(\varepsilon).$$

Proof (i) Fixed $\delta > 0$ and $z \in \mathcal{D}$, by the definition of probability and integration properties we have, for all $j \geq 1$, that

$$\begin{aligned} P\left(X_{k_j} \in (B(z,\delta) \cap \mathcal{D}) \,|\, \sigma(\mathcal{O}_{(k_j)-1})\right) &= \int_{B(z,\delta)\cap\mathcal{D}} g_{k_j}(x \,|\, \sigma(\mathcal{O}_{(k_j)-1}))\, dx \\ &\geq \int_{B(z,\delta)\cap\mathcal{D}} G(x)\, dx. \end{aligned}$$

Using the fact that G is non-negative in $\mathcal{D}$, (6) and Hypothesis H3, we have

$$v(z,\delta) := \int_{B(z,\delta)\cap\mathcal{D}} G(x)\, dx > 0.$$

Thus, for all $j \geq 1$, there exists $v(z,\delta) \in (0,1)$ satisfying (7).

(ii) Given $\varepsilon > 0$, by the continuity of f in x^*, there exists $\delta(\varepsilon) > 0$ such that if $\|x - x^*\| < \delta(\varepsilon)$, then $|f(x) - f(x^*)| < \varepsilon$. Consequently, for each $j \geq 1$ we have the following relation between the events,

$$\begin{aligned} [X_{k_j} \in \mathcal{D} :\, f(X_{k_j}) < f(x^*) + \varepsilon] &= [X_{k_j} \in \mathcal{D} :\, |f(X_{k_j}) - f(x^*)| < \varepsilon] \\ &\supseteq [X_{k_j} \in \mathcal{D} :\, \|X_{k_j} - x^*\| < \delta(\varepsilon)]. \end{aligned}$$

Thus,

$$\begin{aligned} &P\left(X_{k_j} \in \mathcal{D} :\, f(X_{k_j}) < f(x^*) + \varepsilon \,|\, \sigma(\mathcal{O}_{(k_j)-1})\right) \geq \\ &\geq P\left(X_{k_j} \in \mathcal{D} :\, \|X_{k_j} - x^*\| < \delta(\varepsilon) \,|\, \sigma(\mathcal{O}_{(k_j)-1})\right) \geq v(x^*, \delta(\varepsilon)). \end{aligned}$$

Defining $L(\varepsilon) = v(x^*, \delta(\varepsilon)) > 0$, we conclude the proof. □

Consider the set of h-iterations defined by

$$\mathcal{K}_a = \{k \in \mathbb{N} :\, (f(X^*_{k-1}), h(X^*_{k-1})) \text{ is added to the filter}\}. \tag{8}$$

Now we present a sufficient condition that guarantees the convergence of Algorithm 1 in terms of the infimum of the conditional densities of the random vector iterates.

Theorem 2 *Consider the sequence (X_k^*) generated by Algorithm 1. Suppose that there exists a subsequence $(X_{k_j}^*)$ of (X_k^*) such that for each $j \geq 1$, $X_{k_j}^*$ has a conditional density $g_{k_j}(x \,|\, \sigma(\mathcal{O}_{(k_j)-1}))$ satisfying (6). Then, the following statements hold:*

(i) The sequence $(f(X_k^*))$ has an accumulation point that is the global minimum of f over $\mathcal{D}_V$ almost surely.
(ii) If $\mathcal{K}_a$ is finite, then any accumulation point of the sequence $(f(X_k^*))$ is the global minimum of f over $\mathcal{D}_V$ almost surely, this is, $f(X_k^*) \to f^*$ almost surely.
(iii) If x^* is unique, then $X_k^* \to x^*$ almost surely.

Proof (i) Consider the decreasing sequences (ρ_ℓ), (ε_ℓ) of positive terms, converging to zero. By Hypothesis H2 we have that $\mu(T_{\rho_\ell,\varepsilon_\ell}) > 0$, for all $\ell \geq 1$. Thus, the probability that a point randomly generated in $\mathcal{D}$ belongs to $T_{\rho_\ell,\varepsilon_\ell}$ is strictly positive. Using this and the hypothesis about the subsequence $(X_{k_j}^*)$, by Lemma 2, item (i), we have for all $\delta > 0$ and $z \in \mathcal{D}$, there exists $v(z,\delta) \in (0,1)$ satisfying,

$$P\left(X_{k_j}^* \in (B(z,\delta) \cap T_{\rho_\ell,\varepsilon_\ell}) \,|\, \sigma(\mathcal{O}_{(k_j)-1})\right) \geq v(z,\delta). \tag{9}$$

Thus, for all $\ell \geq 1$, there exists k_{j_ℓ} such that $X_{k_{j_\ell}}^* \in T_{\rho_\ell,\varepsilon_\ell}$ almost surely. Defining $S_\varepsilon := \{x \in \mathcal{D} :\, f(x) < f^* + \varepsilon\}$, we have $T_{\rho_\ell,\varepsilon_\ell} \subset S_{\varepsilon_\ell}$ and, consequently, $X_{k_{j_\ell}}^* \in S_{\varepsilon_\ell}$ almost surely.

As the sequence (ε_ℓ) is decreasing and converges to zero, given $\varepsilon > 0$, there exists $\bar{\ell} \in N$, such that $\varepsilon > \varepsilon_{\bar{\ell}} > \ldots > \varepsilon_\ell > \ldots$, for all $\ell \geq \bar{\ell}$, which implies $S_\varepsilon \supseteq S_{\varepsilon_{\bar{\ell}}} \supseteq \ldots \supseteq S_{\varepsilon_\ell} \supseteq \ldots$. Therefore, $X_{k_{j_\ell}}^* \in S_\varepsilon$ for all $\ell \geq \bar{\ell}$. Consequently,

$$X_{k_{j_\ell}}^* \notin S_\varepsilon \Rightarrow X_{k_{j_1}}^* \notin S_\varepsilon, X_{k_{j_2}}^* \notin S_\varepsilon, \cdots X_{k_{j_\ell}}^* \notin S_\varepsilon. \tag{10}$$

Applying Lemma 2, item (ii), there exists $L(\varepsilon) \in (0,1)$ such that for all $j \geq 1$,

$$P\left(X_{k_j}^* \in \mathcal{D} :\, f(X_{k_j}^*) < f(x^*) + \varepsilon \,|\, \sigma(\mathcal{O}_{(k_j)-1})\right) \geq L(\varepsilon). \tag{11}$$

Rewriting (11) in terms of the index k_{j_ℓ} and the set S_ε, we have

$$P\left(X_{k_{j_\ell}}^* \in S_\varepsilon \,|\, \sigma(\mathcal{O}_{(k_{j_\ell})-1})\right) \geq L(\varepsilon). \tag{12}$$

On the other hand,

$$\begin{aligned} P\left(X_{k_{j_1}}^* \notin S_\varepsilon, \ldots, X_{k_{j_\ell}}^* \notin S_\varepsilon\right) &= \prod_{i=1}^{\ell} P\left(X_{k_{j_i}}^* \notin S_\varepsilon \,|\, X_{k_{j_1}}^* \notin S_\varepsilon, \ldots, X_{k_{j_{(i-1)}}}^* \notin S_\varepsilon\right) \\ &= \prod_{i=1}^{\ell} \left(1 - P\left(X_{k_{j_i}}^* \in S_\varepsilon \,|\, X_{k_{j_1}}^* \notin S_\varepsilon, \ldots, X_{k_{j_{(i-1)}}}^* \notin S_\varepsilon\right)\right). \end{aligned} \tag{13}$$

From the conditioning of the random elements in $\mathcal{O}_{k_{j_i}-1}$ and (12),

$$P\left(X_{k_{j_i}}^* \in S_\varepsilon \,|\, X_{k_{j_1}}^* \notin S_\varepsilon, \ldots, X_{k_{j_{(i-1)}}}^* \notin S_\varepsilon\right) \geq L(\varepsilon). \tag{14}$$

Therefore, by (13) and (14), it follows that

$$P\left(X_{k_{j_1}}^* \notin S_\varepsilon, \ldots, X_{k_{j_\ell}}^* \notin S_\varepsilon\right) \leq (1 - L(\varepsilon))^\ell. \tag{15}$$

Thus, for each $\ell \geq 1$, we have

$$0 \leq P\left(f\left(X^*_{k_{j_\ell}}\right) - f^* \geq \varepsilon\right) = P\left(f\left(X^*_{k_{j_\ell}}\right) \geq f^* + \varepsilon\right) = P\left(X^*_{k_{j_\ell}} \notin S_\varepsilon\right) \leq$$
$$\leq P\left(X^*_{k_{j_1}} \notin S_\varepsilon, \ldots, X^*_{k_{j_\ell}} \notin S_\varepsilon\right) \leq (1 - L(\varepsilon))^\ell.$$

Consequently,

$$\lim_{\ell \to \infty} P\left(f\left(X^*_{k_{j_\ell}}\right) - f^* \geq \varepsilon\right) = 0,$$

which means that $f\left(X^*_{k_{j_\ell}}\right) \to f^*$ in probability. Hence, by a standard result in probability theory [[40], Thm 6.3.1(b)] we have that $f(X^*_{k_{j_{\ell(i)}}}) \to f^*$ almost surely, with $i \to \infty$ for some subsequence of index $(k_{j_{\ell(i)}})_{i \geq 1}$.

(ii) Now, suppose that $\mathcal{K}_a$ is finite. By the filter update rule, the number of iterations in which the objective function increases is finite. Consequently, there exists $\bar{k} \in \mathbb{N}$ such that the sequence $(f(X^*_k))_{k \geq \bar{k}}$ is non-increasing. On the other hand, given that $f^* > -\infty$, we have $\lim_{k \to \infty} f(X^*_k(\omega))$ exists for each underlying sample point. Thus, by the uniqueness of the limit, we can conclude that

$$f(X^*_k) \to f^* \quad \text{a.s. (almost surely)} \tag{16}$$

(iii) Given $\varepsilon > 0$, define $\tilde{f} := \inf_{x \in \mathcal{D}, \|x - x^*\| \geq \varepsilon} f(x)$. By the uniqueness of x^*, it follows $\tilde{f} > f^*$. From (16), we have that such convergence occurs less than in a set of Lebesgue measure zero; i.e, there exists $\mathcal{N} \subset \Omega$ with $P(\mathcal{N}) = 0$ and such that $f(X^*_k(\omega)) \to f(x^*)$ for all $\omega \in \mathcal{N}^c$. Thus, for all $\omega \in \mathcal{N}^c$ there exists $N \in \mathbb{N}$ such that for all $k \geq N$,

$$f(X^*_k(\omega)) - f(x^*) = |f(X^*_k(\omega)) - f(x^*)| < \tilde{f} - f(x^*),$$

which implies,

$$f(X^*_k(\omega)) < \tilde{f}, \tag{17}$$

for all $k \geq N$ and $\omega \in \mathcal{N}^c$.

Now, let us show that $\|X^*_k(\omega) - x^*\| < \varepsilon$, for all $\omega \in \mathcal{N}^c$. In fact, suppose by contradiction that there exists $\omega \in \mathcal{N}^c$ such that $\|X^*_k(\omega) - x^*\| \geq \varepsilon$. By the definition of $\tilde{f}$, it follows that $f(X^*_k(\omega)) \geq \tilde{f}$, which contradicts (17). Hence, we have that for all $\omega \in \mathcal{N}^c$, there exists $N \in \mathbb{N}$ such that, for all $k \geq N$, $\|X^*_k(\omega) - x^*\| < \varepsilon$. Therefore, $X^*_k(\omega) \to x^*$ a.s. □

Next theorem ensures the convergence in probability of Algorithm 1 when a subsequence of the iterates is given as the sum of the best point obtained so far with random vectors with multivariate normal distribution.

Theorem 3 *Consider the sequence (X^*_k) generated by Algorithm 1 and suppose that there exists a subsequence $(X^*_{k_j})_{j \geq 1}$ such that, for each $j \geq 1$,*

$$X^*_{k_j} = X^*_{(k_j)-1} + W_j, \tag{18}$$

where W_j is a random vector which conditional distribution $\sigma(\mathcal{O}_{(k_j)-1})$ is a multivariate normal distribution with mean vector 0 and covariance matrix V_j. For each $j \geq 1$, let $\bar{\lambda}_j$ be the smallest eigenvalue of V_j. If $\inf_{j \geq 1} \bar{\lambda}_j > 0$, then the sequence $f(X^*_k)$ has an

accumulation point that is the global minimum of f over $\mathcal{D}_V$ almost surely. In addition, if $\mathcal{K}_a$ is finite, then $f(X_k^*) \to f^*$ almost surely. Finally, if x^* is the unique global minimizer, then $X_k^* \to x^*$ almost surely.

Proof By the mechanism of the algorithm,

$$X^*_{(k_j)-1} = \sum_{i=1}^{(k_j)-1} X_i 1_E(X_i), \tag{19}$$

where 1_E is the indicator function and E is the event defined by

$$\begin{aligned} E = &[h(X_\ell) < h(X_t) + \varepsilon_h, \text{ for all } t = 1, \ldots, (k_j) - 1 \text{ and } \ell \text{ is the smallest index} \\ &\text{of the points non-dominated by the filter } \bar{\mathcal{F}}_t \text{ with this property}]. \end{aligned}$$

For each $i = 1, \ldots, (k_j) - 1$, X_i is obtained by a deterministic function of the random elements in $\mathcal{O}_i$. Consequently, $X^*_{(k_j)-1}$ is obtained by a deterministic function of the random elements in $\mathcal{O}_{(k_j)-1}$. By (18), the conditional distribution of X_{k_j}, given $\sigma(\mathcal{O}_{(k_j)-1})$, is multivariate normal distribution, with mean vector $X^*_{(k_j)-1}$ and covariance matrix V_j. This matrix is invertible, for all $j \geq 1$, since by hypothesis $\inf_{j\geq 1} \bar{\lambda}_j > 0$. Thus, X_{k_j} has conditional density given by

$$g_{k_j}(x \,|\, \sigma(\mathcal{O}_{(k_j)-1})) = \gamma[det(V_j)]^{-1/2} \Psi\left((x - X^*_{(k_j)-1})^T V_j^{-1} (x - X^*_{(k_j)-1})\right), \tag{20}$$

where $\Psi(x) = e^{-x/2}$ and $\gamma = (2\pi)^{-n/2}$.

From the definition of the Euclidean norm and of $\bar{\lambda}_j$, it follows that

$$\|V_j^{-1}\| = \sqrt{\text{largest eigenvalue of } \left(V_j^{-1}\right)^T V_j^{-1}} = \sqrt{\frac{1}{\bar{\lambda}_j^2}} = \frac{1}{|\bar{\lambda}_j|} = \frac{1}{\bar{\lambda}_j}.$$

Using this, the fact that V_j is definite positive and the Cauchy-Schwarz inequality, we have, for each $j \geq 1$ and $x \in \mathcal{D}$,

$$(x - X^*_{(k_j)-1})^T V_j^{-1} (x - X^*_{(k_j)-1}) \leq \|x - X^*_{(k_j)-1}\|^2 \|V_j^{-1}\| \leq \frac{D^2}{\bar{\lambda}_j},$$

where $D = diam(\mathcal{D}) = \sup_{x,y\in\mathcal{D}} \|x - y\| < \infty$, once that $\mathcal{D}$ is bounded. As Ψ is monotonically non-increasing,

$$\Psi\left((x - X^*_{(k_j)-1})^T V_j^{-1} (x - X^*_{(k_j)-1})\right) \geq \Psi\left(\frac{D^2}{\bar{\lambda}_j}\right). \tag{21}$$

Thus, $det(V_j) \leq \left(\lambda_j^*\right)^n$, where λ_j^* is the largest eigenvalue of V_j. Using this and (21) in (20) and the fact that Ψ is monotonically non-increasing we have, for each $x \in \mathcal{D}$,

$$g_{k_j}(x \,|\, \sigma(\mathcal{O}_{(k_j)-1})) \geq \gamma \left(\lambda_j^*\right)^{-n/2} \Psi\left(\frac{D^2}{\bar{\lambda}_j}\right) \geq \gamma \left(\sup_{\ell\geq 1} \lambda_\ell^*\right)^{-n/2} \Psi\left(\frac{D^2}{\inf_{\ell\geq 1} \bar{\lambda}_\ell}\right).$$

Thus, for all $x \in \mathcal{D}$,

$$G(x) = \inf_{j \geq 1} g_{k_j}(x \,|\, \sigma(\mathcal{O}_{(k_j)-1})) \geq \gamma \left(\sup_{\ell \geq 1} \lambda_\ell^* \right)^{-n/2} \Psi \left(\frac{D^2}{\inf_{\ell \geq 1} \bar{\lambda}_\ell} \right) > 0,$$

which implies

$$\mu\left(\{x \in \mathcal{D} : \, G(x) = 0\}\right) = 0.$$

Consequently, the hypotheses of Theorem 2 hold and we get the results. □

Therefore, any filter-based stochastic algorithm with the same general structure as Algorithm 1, satisfying the sufficient conditions studied in this section almost surely converges to a solution of the problem.

3 The algorithm with surrogate models

In this section, we specify the steps of Algorithm 1 considering Radial Basis Function surrogate models to approximate the expensive objective function of problem (1). We refer the resulting algorithm as FbSA-RBF.

The iterates are computed by a dynamic coordinate search strategy similar to the one used in DDS [45]. For an efficient exploration of the search space, at each iteration k, n_t random trial points are generated by adding random perturbations with normal distribution in the dynamically chosen coordinates of the best current point. Then, the best trial point among the n_t trial points is selected, that can replace or not, the current best point. In order to reduce the number of function evaluations, a radial basis function interpolation model is used to predict the objective function evaluation in the trial points.

3.1 Description of the algorithm

This section describes the main choices and steps of FbSA-RBF algorithm.

3.1.1 Initial points

The algorithm computes a set of $n_0 \geq n + 1$ initial points:

$$I = \{(X_j) \in \mathcal{D} : \, X_j = \ell + \beta_j (u - \ell), \quad \text{with} \quad 1 \leq j \leq n_0\}, \tag{22}$$

where $\beta_j \in [0,1]^n$ is generated by using a symmetric Latin hypercube design, as proposed in [49]. Then, the best initial point $X_0^* \in I$ is selected as follows.

Definition 1 The initial point X_0^* is the point of the set I which satisfies one of the following two conditions:

(i) if all points in I are infeasible, then the best initial point will be the one with the smallest infeasibility measure. In case of a tie, select the point with the smallest objective function value;
(ii) if the set I contains feasible points, then the best initial point will be the feasible trial point with the smallest objective function value.

3.1.2 Trial points

At each iteration k, the algorithm generates n_t trial points by perturbing some or all components of the current best point. The collection of random elements used for computing the trial points is given by

$$\{\Lambda_{k_j}\}_{1\leq j\leq 3n_t+1} = \{P_k, Z_k^t, \xi_k^t, i_k^t\}, \tag{23}$$

where P_k is the probability of perturbing the coordinates; Z_k^t is the vector associated to t-th trial point, whose components are random numbers uniformly distributed in the interval $[0,1]$, and that with the probability P_k will decide to perturb or not certain component of the best current point; $(\xi_k^t)_i \sim N(0,(\lambda_{k-1})_i^2)$ is related to the step size as a perturbation to the component i of the current best point, chosen to be perturbed, with a normal distribution with mean 0 and standard deviation $(\lambda_{k-1})_i$ and i_k^t is the index associated to the component that can be perturbed.

For each $t = 1,\ldots,n_t$, we define the set of indices of the coordinates that will be perturbed at the iteration k, by

$$J_k^t = \begin{cases} \{i_k^t\}, & \text{if } (Z_k^t)_i > P_k, \text{ for all } i = 1,\ldots,n \\ \{i : (Z_k^t)_i \leq P_k\}, & \text{otherwise.} \end{cases} \tag{24}$$

The step size $(\xi_k^t)_i$ is adjusted by controlling the standard deviation λ_k, associated with the magnitude of the step perturbation. As this parameter is related to the control of the randomness or, equivalently, of the diversity of the points generated, it can be increased or decreased depending of the number of success or unsuccessful consecutive iterations.

The deterministic function $\Theta^t : \mathcal{O}_k \to \mathcal{D}$ computes the trial points as a perturbation of the current best point as

$$(X_k^t)_i = \begin{cases} \min\{u_i, \max\{(X_{k-1}^*)_i + (\xi_k^t)_i, \ell_i\}\}, & \text{if } i \in J_k^t \\ (X_{k-1}^*)_i, & \text{if } i \notin J_k^t, \end{cases} \tag{25}$$

for all $t = 1,\ldots,n_t$ and $i = 1,\ldots,n$.

3.1.3 Radial basis functions

Surrogate models are computationally inexpensive approximations that are used to reduce the high cost of the objective function evaluations involved in the optimization process [7,39]. Since f is expensive, we use a surrogate model to approximate f across a RBF model [33,39], briefly described below.

Consider the set $\mathcal{A}_{k-1} = \{X_1,\ldots,X_q\} \subset \mathbb{R}^n$ of sample points where the values of the objective function $f(X_1),\ldots,f(X_q)$ are known. The RBF model approximates the objective function through the following interpolation function,

$$S_k(X) = \sum_{X_i \in \mathcal{A}_{k-1}} \eta_i \, \phi(\|X - X_i\|) + p(X), \tag{26}$$

where $\eta_i \in \mathbb{R}$, for $i = 1,\ldots,q$, $\|\cdot\|$ is the Euclidean norm, p is a linear function in $\mathbb{R}^n$ and the function ϕ has the cubic form, $\phi(y) = y^3$, as suggested in [39]. Note

that other functions can be used, for example $\phi(y) = y^2 \ln(y)$ or the Gaussian $\phi(y) = e^{-\gamma y^2}$ with $\gamma > 0$, as proposed in [33].

Consider $\Phi \in \mathbb{R}^{q\times q}$ the matrix defined by $\Phi_{(i,j)} := \phi(\|X_i - X_j\|)$, for $i, j = 1, \ldots, q$, and $M_P \in \mathbb{R}^{q\times(n+1)}$ the matrix whose i-th line is given by $[1 \quad X_i^T]$. Thus, the cubic RBF model that interpolates the points $(X_1, f(X_1)), \ldots, (X_q, f(X_q))$ is obtained by solving the system,

$$\begin{pmatrix} \Phi & M_P \\ M_P^T & 0 \end{pmatrix} \begin{pmatrix} \eta \\ c \end{pmatrix} = \begin{pmatrix} F \\ 0_{n+1} \end{pmatrix}, \tag{27}$$

where $F = (f(X_1), \ldots, f(X_q))$, $\eta = (\eta_1, \ldots, \eta_q)^T \in \mathbb{R}^q$ and $c = (c_1, \ldots, c_{n+1}) \in \mathbb{R}^{n+1}$ are the coefficients of the linear function p. The coefficient matrix in (27) is invertible if and only if $\text{rank}(M_P) = n + 1$, which is equivalent to having a subset of $n+1$ affinely independent points among the sample set $\mathcal{A}_{k-1}$ [30,33,36]. Along the iterative process, the best points are added to the sample set to update the RBF model ensuring that the M_P matrix will always have rank $n + 1$.

3.1.4 Best trial point selection

The RBF model S_k and the infeasibility measure h are used to select the best trial point $X_k^{t\,*}$ from the set of trial points X_k^t, for $t = 1, \ldots, n_t$. The concept of non-dominance (4) is considered, replacing (f^j, h^j) by $(S_k(X_k^t), h(X_k^t))$. So, a point Y, or equivalently the pair $(S_k(Y), h(Y))$, is non-dominated by the pair $(S_k(X_k^t), h(X_k^t))$, if

$$S_k(Y) + \alpha h(Y) \leq S_k(X_k^t) \quad \text{or} \quad h(Y) \leq (1-\alpha) h(X_k^t)). \tag{28}$$

Based on this, the best trial point is selected according to the following definition.

Definition 2 The best trial point $X_k^{t\,*}$ at the iteration k is the trial point non-dominated by any other pair $(S_k(X_k^t), h(X_k^t))$, according to (28), with the smallest value of the infeasibility measure h.

After the selection of the best trial point $X_k^{t\,*}$, the algorithm verifies if such point is accepted by the current filter. So, the value $f(X_k^{t\,*})$ is computed. If $X_k^{t\,*} \notin \bar{\mathcal{F}}_{k-1}$, it is accepted as the new iterate X_k and it is included to the set $\mathcal{A}_k$ of the points used by the RBF model for the next iteration. Otherwise, the iteration is unsuccessful and we keep the best current point from the previous iteration.

3.1.5 The algorithm

Now we present the Filter-based stochastic algorithm FbSA assisted by Radial Basis Function surrogate models to approximate the objective function.

Algorithm 2 - FbSA-RBF

Require: $\alpha, \eta \in (0,1)$, λ_0, $\varepsilon_h > 0$, τ_{suc}, $\tau_{failure}$, $n_0 \geq n+1$, n_t, $N_{max} \in \mathbb{N}$

generate the set I of n_0 initial points according to (22)
select $X_0^* \in I$ by Definition 1
set $\mathcal{O}_0 = \{X_0^*\}$, $F_0 = \mathcal{F}_0 = \emptyset$, $\mathcal{A}_0$=I, $c_r = c_{suc} = c_{failure} = 0$ and $n_f = n_0$,
set $k = 1$
while the stopping criterion is not satisfied **do**
  set $\bar{F}_{k-1} = F_{k-1} \cup \{(f(X_{k-1}^*), h(X_{k-1}^*))\}$ and define
  $\bar{\mathcal{F}}_{k-1} = \mathcal{F}_{k-1} \cup \left\{x \in \mathcal{D} : \ f(x) + \alpha h(x) \geq f(X_{k-1}^*) \ \text{ and } \ h(x) \geq (1-\alpha)h(X_{k-1}^*)\right\}$
  build/update the RBF model S_k by (26) and (27) using the points of $\mathcal{A}_{k-1}$
  generate a collection of random elements $\{\Lambda_{k_j}\}_{1 \leq j \leq 3n_t+1} = \{P_k, Z_k^t, \xi_k^t, i^t\}$, with $t = 1, \ldots, n_t$, according to (23)
  set $\mathcal{O}_k = \mathcal{O}_{k-1} \cup \{\Lambda_{k_j}\}_{1 \leq j \leq 3n_t+1}$
  compute the set J_k^t by (24), for $t = 1, \ldots, n_t$
  compute $X_k^t \in \mathcal{D}$ by (25), for $t = 1, \ldots, n_t$
  select $X_k^{t\,*}$ by Definition 2, compute $f(X_k^{t\,*})$ and set $n_f = n_f + 1$
  if $X_k^{t\,*} \notin \bar{\mathcal{F}}_{k-1}$ **then**
    set $X_k = X_k^{t\,*}$, $\mathcal{A}_k = \mathcal{A}_{k-1} \cup \{X_k^{t\,*}\}$
    Filter update:
    if $f(X_k) \leq f(X_{k-1}^*)$ **then**
      set $F_k = F_{k-1}$ and $\mathcal{F}_k = \mathcal{F}_{k-1}$ (f-iteration)
    else
      set $F_k = \bar{F}_{k-1}$ and $\mathcal{F}_k = \bar{\mathcal{F}}_{k-1}$ (h-iteration)
    end if
    Best point update:
    if $h(X_k) < h(X_{k-1}^*) + \varepsilon_h$ **then**
      set $X_k^* = X_k^{t\,*}$ (successful iteration)
      if $\left(f(X_{k-1}^*) - f(X_k^{t\,*})\right) > \eta |f(X_{k-1}^*)|$ **then**
        set $c_{suc} = c_{suc} + 1$ and $c_{failure} = 0$
      end if
    else
      set $X_k^* = X_{k-1}^*$, $c_{failure} = c_{failure} + 1$ and $c_{suc} = 0$ (unsuccessful iteration)
    end if
  else
    set $X_k = X_{k-1}^*$, $X_k^* = X_k$, $\mathcal{A}_k = \mathcal{A}_{k-1}$, $F_k = F_{k-1}$, $\mathcal{F}_k = \mathcal{F}_{k-1}$
  end if
  Adjust step size:
  if $c_{failure} \geq \tau_{failure}$ **then**
    set $\lambda_k = \max(\lambda_{k-1}/2, \lambda_0)$, $c_r = c_r + 1$ and $c_{failure} = 0$
  else
    if $c_{suc} \geq \tau_{suc}$ **then**
      set $\lambda_k = \min(2\lambda_{k-1}, \lambda_0)$ and $c_{suc} = 0$
    end if
  end if
  set $k = k + 1$
end while

3.2 Convergence analysis

This section is dedicated to prove that Algorithm 2 satisfies the sufficient conditions established in Theorems 1 and 3, ensuring its almost surely convergence.

In FbSA-RBF algorithm, the choice of the coordinates to be perturbed follows a uniform distribution, while the magnitude of perturbation follows a normal distribution. We will prove that the sequence (X_k^*) generated by the algorithm admits

a subsequence where all components have a random perturbation with normal distribution. For that, next lemma shows that the existence of such subsequence is ensured when the probability $P_k \in (0,1]$ is greater than a positive constant.

Lemma 3 *Consider the sequence (X_k^*) generated by Algorithm 2 and assume that there exists a constant $\overline{p} > 0$ such that the probability $P_k \geq \overline{p}$, for all $k \in \mathbb{N}$. Then, there exists a subsequence $\left(X_{k_j}^*\right)$ in $\mathcal{D}$ where all components of all of its terms have random perturbations with normal distribution. More precisely,*

$$\left(X_{k_j}^*\right)_i = \min\{u_i, \max\{(X_{(k_j)-1}^*)_i + (\xi_{k_j})_i,\, \ell_i\}\}, \tag{29}$$

with $(\xi_{k_j})_i \sim N(0, (\lambda_{k_j})_i^2)$, for $i = 1, \ldots, n$.

Proof Consider the success event E_k as the one in which all the coordinates of X_k^* have a random perturbation with normal distribution. In this case, according to (24),

$$[E_k] = [(Z_k)_i \leq P_k, \quad \text{for all} \quad i = 1, \ldots, n],$$

where $(Z_k)_i \sim U[0,1]$. Since the events $[(Z_k)_1 \leq P_k]$, ..., $[(Z_k)_n \leq P_k]$ are independent, we have

$$P(E_k) = \prod_{i=1}^{n} P\left((Z_k)_i \leq P_k\right). \tag{30}$$

As each $(Z_k)_i \sim U[0,1]$ we have that the density function is given by

$$g(z) = \begin{cases} 1, \text{ if } z \in [0,1], \\ 0, \text{ otherwise.} \end{cases}$$

Thus, for each i,

$$P\left((Z_k)_i \leq P_k\right) = \int_{-\infty}^{P_k} g(z)\, dz = \int_{-\infty}^{0} 0\, dz + \int_{0}^{P_k} 1\, dz = P_k.$$

Substituting in (30), we have

$$P(E_k) = \prod_{i=1}^{n} P_k = P_k^n.$$

By the hypothesis that $P_k \geq \overline{p} > 0$, it follows

$$\sum_{k=1}^{\infty} P(E_k) = \sum_{k=1}^{\infty} P_k^n \geq \sum_{k=1}^{\infty} \overline{p}^{\,n} = \overline{p}^{\,n} \sum_{k=1}^{\infty} 1 = \infty.$$

Therefore, by Lemma of Borel-Cantelli [40],

$$P\left(E_k \text{ infinite times}\right) = 1.$$

Thus, the sequence (X_k^*) admits a subsequence $\left(X_{k_j}^*\right)$ such that all components of each element of the sequence are perturbed. By the algorithm, the terms of the subsequence $\left(X_{k_j}\right)$ are computed, coordinates to coordinates, by

$$\left(X_{k_j}^*\right)_i = \min\{u_i, \max\{(X_{(k_j)-1}^*)_i + (\xi_{k_j})_i,\, \ell_i\}\},$$

with $(\xi_{k_j})_i \sim N(0, (\lambda_{k_j})_i^2)$, for $i = 1, \ldots, n$, which completes the proof. □

From last lemma, there exists a subsequence $(X_{k_j}^*)$, of the sequence (X_k^*) generated by Algorithm 2, satisfying (18), which implies the convergence of the algorithm almost surely as shown in next theorem.

Theorem 4 *Assume that Hypotheses H1 - H3 hold. Consider (X_k^*) a sequence generated by Algorithm 2. Suppose that there exists $\overline{p} > 0$ such that with $P_k \geq \overline{p}$, for all $k \in \mathbb{N}$. Then, the sequence $f(X_k^*)$ has an accumulation point that is global minimum of the f over $\mathcal{D}_V$ almost surely. In addition, if $\mathcal{K}_a$ is finite, then $f(X_k^*) \to f^*$ almost surely. Finally, if x^* is the unique global minimizer of f, then $X_k^* \to x^*$ almost surely.*

Proof By Lemma 3, there exists a subsequence $\left(X_{k_j}^*\right)$ where all coordinates of each element have random perturbation with normal distribution, with $(X_{k_j}^*)_i$ given by (29), where each component $(\xi_{k_j})_i \sim N(0, (\lambda_{k_j})_i^2)$ follows a normal distribution of mean 0 and standard deviation given by the square root of the eigenvalues of the covariance matrix. Thus, $\xi_{k_j} \sim N(0, V_j)$ is a random vector whose conditional distribution, given $\sigma(\mathcal{O}_{(k_j)-1})$, is the multivariate normal distribution with mean 0 and covariance matrix V_j given by

$$V_j = Cov(\xi_{k_j}) = diag\left((\lambda_{k_j})_1^2, \ldots, (\lambda_{k_j})_n^2\right).$$

In this case, we have that the eigenvalues of V_j are the variance $(\lambda_{k_j})_1^2$, ..., $(\lambda_{k_j})_n^2$ of the normal random perturbation added to each component of $X_{(k_j)-1}^*$. Thus, the smallest eigenvalue of V_j is given by

$$\bar{\lambda}_j := \min_{1 \leq i \leq n} (\lambda_{k_j})_i^2 > 0.$$

Consequently, the results follow from Theorem 3. □

4 Numerical experiments

In this section, we report the numerical experiments to illustrate the practical performance of the proposed FbSA-RBF when solving three benchmark sets of constrained global problems.

First, a comparison is carried out with other methods in the literature, which involve different approaches for handling the constraints, namely the A2-SF, P-BS AFS and EPGO, presented in [12,28,42], using the set of problems described in [6]. Second, well-known engineering design problems presented in [43] are used in order to compare to A2-SF algorithm. In the third experiment, a set of problems described in [26,29] is used to compare the performance of FbSA-RBF with CARS-RBF, CPRS-RBF and ConstrLMSRBF algorithms [30,36], which involve RBF surrogate models.

4.1 Implementation details

In the two first numerical experiments, stopping conditions based on the values of h and f are used. Thus, the proposed FbSA-RBF algorithm stops when the best current point X_k^* satisfies

$$f(X_k^*) \leq f^* + \varepsilon_f \quad \text{and} \quad h(X_k^*) \leq \varepsilon_h, \tag{31}$$

where ε_f is the accuracy error required in the value of the objective function, ε_h is the tolerance of the constraint violation and f^* is the best-known solution available in the literature. Besides, if (31) does not hold, the FbSA-RBF algorithm stops if it reaches the maximum number of function evaluations $N_{max} = 1000$ allowed. The accuracy required for the objective function and constraint violation were the same as those used in [28,42], $\varepsilon_f = 10^{-4}$ and $\varepsilon_h = 10^{-8}$, respectively.

The probability of perturbing the coordinates P_k is computed by

$$P_k = \left(1 - \frac{\log(n_f - n_0 + 1)}{\log(N_{max} - n_0)}\right) \min\left\{\frac{20}{n}, 1\right\},$$

where n is the dimension of the problem, $n_0 = 2(n+1)$ is the number of initial points, n_f is the number of function evaluations until the iteration k. As the algorithm evaluates the objective function once per iteration, as k increases, the probability tends to decrease. This is reasonable since at the end of the iterative process it is expected that the best current point is close to the solution of the problem and therefore major perturbations are not need.

The number of trial points n_t generated at each iteration varies according to the dimension of the problem. The values used in the numerical experiments of FbSA-RBF were $n_t = \min\{100n, 5000\}$, the maximum tolerance of failures, $\tau_{failure} = \max\{5, n\}$, and the maximum tolerance of success, $\tau_{suc} = 3$, as suggested in [39].

Since the algorithm relies on some random parameters and variables, each problem was solved 30 times. Problems with equality constraints were reformulated to inequality constrained problems, by considering $|c_i(x)| \leq 10^{-4}$, for $i \in \mathcal{E}$. The tests were performed in a notebook ASUSTek Intel Core i7-6700HQ, CPU 2.60GHz, with 16GB RAM, 64-bit, using Matlab R2017a.

4.2 Numerical results

In the first experiment, a set of 20 global optimization test problems with general constraints described in[6] is used to illustrate the practical performance of FbSA-RBF. The number of variables ranges from 2 to 10 and the number of constraints ranges from 1 to 12. The results of this study are compared to A2-SF, P-BS AFS and EPGO.

The A2-SF [28] combines the filter methodology, with a slanting filter rule, and the DDS method. The P-BS AFS [42] is a population-based filter framework, where the subproblems are globally solved by an artificial fish swarm stochastic algorithm. The EPGO [12], uses an exact penalty approach based on the DIRECT to globally solve the subproblems.

Table 1 presents the numerical results of the FbSA-RBF, A2-SF, P-BF AFS and EPGO for this set of problems. The first three columns display the data of the problem: the identification (P); the dimension (n); the global optimum known (f^*). The next columns display, for each algorithm, the results obtained among the 30 runs: the best solution obtained (f_{best}), the median (f_{med}), the infeasibility measure at the best point (h_{best}), the median of infeasibility measure (h_{med}) and the average number of function evaluations ($n_{f_{avg}}$). The numerical results of the A2-SF, P-BF AFS and EPGO were taken from [28], [42] and [12], respectively.

Table 1: Numerical results for problems described in [6]

P	n	f^*	Algorithm	f_{best}	f_{med}	h_{best}	h_{med}	$n_{f_{avg}}$
1	5	2.9313e-02	FbSA-RBF	0.2724	141.6525	0.00e00	0.00e00	1000
			A2-SF	0.1978	237.3387	1.43e-04	6.49e-05	9479
			P-BF AFS	0.0956	1.4665	7.84e-07	*	6945
			EPGO	0.0625		2.35e-07		39575
2a	9	-4.0000e02	FbSA-RBF	-388.622	0.000	0.00e00	0.00e00	1000
			A2-SF	-395.875	-312.487	0.00e00	0.00e00	14994
			P-BF AFS	-358.650	-308.664	0.00e00	*	7068
			EPGO	-134.113		8.43e-04		115107
2b	9	-6.0000e02	FbSA-RBF	-284.276	986.439	0.00e00	0.00e00	1000
			A2-SF	-384.423	-301.721	0.00e00	0.00e00	14743
			P-BF AFS	-378.317	-274.472	0.00e00	*	6963
			EPGO	-768.457		5.30e-04		120057
2c	9	-7.5000e02	FbSA-RBF	-684.722	-544.883	0.00e00	0.00e00	1000
			A2-SF	-747.021	-702.915	0.00e00	0.00e00	14630
			P-BF AFS	-697.452	-657.349	0.00e00	*	7189
			EPGO	-82.977		8.43e-04		102015
2d	10	-4.0000e02	FbSA-RBF	-400.000	-50.386	0.00e00	0.00e00	1000
			A2-SF	-399.900	-347.957	0.00e00	0.00e00	14286
			P-BF AFS	-399.118	-394.563	0.00e00	*	6526
			EPGO	-385.170		0.00e00		229773
3a	6	-3.8880e-01	FbSA-RBF	-0.3889	-0.3795	0.00e00	0.00e00	1000
			A2-SF	-0.3878	-0.3747	0.00e00	0.00e00	15858
			P-BF AFS	-0.3888	-0.3842	5.22e-04	*	7495
			EPGO	-0.3861		1.02e-06		48647
3b	2	-3.8881e-01	FbSA-RBF	-0.3888	-0.3885	0.00e00	0.00e00	907
			A2-SF	-0.3888	-0.3883	0.00e00	0.00e00	6512
			P-BF AFS	-0.3888	-0.3888	0.00e00	*	1041
			EPGO	-0.3888		0.00e00		3449
4	2	-6.6666e00	FbSA-RBF	-6.6666	-6.6665	0.00e00	0.00e00	361
			A2-SF	-6.6666	-6.6662	0.00e00	0.00e00	5483
			P-BF AFS	-6.6667	-6.6665	0.00e00	*	493
			EPGO	-6.6666		0.00e00		3547
5	3	2.0116e02	FbSA-RBF	201.159	201.159	0.00e00	0.00e00	1000
			A2-SF	201.159	201.157	0.00e00	3.57e-02	2930
			P-BF AFS	201.159	201.159	8.11e-07	*	2999
			EPGO	201.159		1.66e-04		14087
6	2	3.7629e02	FbSA-RBF	376.292	376.293	0.00e00	0.00e00	1000
			A2-SF	376.305	376.986	0.00e00	0.00e00	6079
			P-BF AFS	376.293	376.304	0.00e00	*	1335
			EPGO	0.470		2.05e-05		1523
7	2	-2.8284e00	FbSA-RBF	-2.8284	-2.8283	0.00e00	0.00e00	194
			A2-SF	-2.8284	-2.8230	0.00e00	0.00e00	4829
			P-BF AFS	-2.8284	-2.8283	0.00e00	*	920
			EPGO	-2.8058		0.00e00		13187

Continued on next page

Table 1 *(Continued from previous page)*

P	n	f^*	Algorithm	f_{best}	f_{med}	h_{best}	h_{med}	$n_{f_{avg}}$
8	2	-1.1870e02	FbSA-RBF	-118.704	-86.422	0.00e00	0.00e00	1000
			A2-SF	-118.703	-115.138	0.00e00	0.00e00	5904
			P-BF AFS	-118.704	-118.698	0.00e00	*	1521
			EPGO	-118.704		0.00e00		7621
9	6	-1.3402e01	FbSA-RBF	-13.4019	-13.4019	0.00e00	0.00e00	1000
			A2-SF	-13.4019	-13.3916	0.00e00	0.00e00	8187
			P-BF AFS	-13.4018	-13.4007	0.00e00	*	1839
			EPGO	-13.4026		1.35e-04		68177
10	2	7.4178e-01	FbSA-RBF	0.7418	0.7418	0.00e00	0.00e00	68
			A2-SF	0.7419	0.7436	0.00e00	0.00e00	5733
			P-BF AFS	0.7418	0.7418	0.00e00	*	2126
			EPGO	0.7420		0.00e00		6739
11	2	-5.0000e-01	FbSA-RBF	-0.5000	-0.4999	0.00e00	0.00e00	80
			A2-SF	-0.5000	-0.4982	0.00e00	0.00e00	6135
			P-BF AFS	-0.5000	-0.5000	0.00e00	*	782
			EPGO	-0.5000		0.00e00		3579
12	2	-1.6739e01	FbSA-RBF	-16.7393	-16.7321	0.00e00	0.00e00	1000
			A2-FF	-16.6486	-15.5805	0.00e00	3.02e-05	4159
			P-BF AFS	-16.7389	-16.7389	0.00e00	*	35
			EPGO	-16.7389		5.36e-06		3499
13	3	1.8935e02	FbSA-RBF	271.994	287.201	0.00e00	1.17e-04	1000
			A2-SF	278.942	280.580	1.45e-05	3.30e-01	4601
			P-BF AFS	189.345	253.937	0.00e00	*	4031
			EPGO	195.955		9.21e-04		8085
14	4	-4.5142e00	FbSA-RBF	-4.5142	-4.5142	0.00e00	0.00e00	523
			A2-SF	-4.5142	-4.4808	0.00e00	0.00e00	8520
			P-BF AFS	-4.5142	-4.5139	0.00e00	*	2028
			EPGO	-4.3460		9.22e-05		19685
15	3	0.0000e00	FbSA-RBF	0.0000	0.0000	0.00e00	1.25e-02	1000
			A2-SF	0.0000	0.0000	2.03e-05	1.91e-02	4729
			P-BF AFS	0.0000	0.0000	9.11e-07	*	3593
			EPGO	0.0000		4.94e-05		1645
16	5	7.0492e-01	FbSA-RBF	0.7050	0.7050	0.00e00	0.00e00	21
			A2-SF	0.7049	0.7050	0.00e00	0.00e00	121
			P-BF AFS	0.7049	0.7049	0.00e00	*	447
			EPGO	0.7181		2.00e-04		22593

*information not available at source

From Table 1, we can conclude that the proposed algorithm presents competitive results when compared with the P-BF AFS, A2-SF and EPGO algorithms. Using less function evaluations, FbSA-RBF reaches the global optimum value known f^* on 15 out of 20 problems. Furthermore, the median objective function values are close to f^* for 12 of them. Four of the problems (1, 2a, 2b and 2c) are not solved by any algorithm, although some of them obtained feasible solutions.

The second experiment involves nine more complex and real application engineering design problems described in [43], since problems of practical interest are important for assessing the effectiveness of the proposed algorithm. In this set of problems, the number of design variables ranges from 2 to 8 and the number of inequality constraints ranges from 1 to 11.

Table 2 presents the results obtained by FbSA-RBF and A2-SF [28]. The columns related to the infeasibility measures are omitted since both algorithms

produced feasible solutions in all runs. Using less function evaluations, FbSA-RBF reaches best solutions in all problems, comparing to A2-SF. None algorithm solved with a good accuracy the problems C. Vessel, Heat E. and W. Beam.

Table 2 Numerical results for engineering design problems described in [43]

P	n	f^*	Algorithm	f_{best}	f_{med}	$n_{f_{avg}}$
C.Vessel	5	5868.7650	FbSA-RBF	5885.4030	6263.9640	1000
			A2-SF	5898.3626	6327.0383	10966
D.Brake	4	0.1274	FbSA-RBF	0.1274	0.1274	793
			A2-SF	0.1274	0.1283	6770
F.B.Truss	4	1400.000	FbSA-RBF	1400.0000	1400.0000	14
			A2-SF	1400.0000	1400.0000	336
Heat E.	8	7049.2480	FbSA-RBF	7051.1860	7666.1930	1000
			A2-SF	7075.0293	8340.3915	17826
S.Reducer	7	2994.4991	FbSA-RBF	2994.4840	3089.5885	1000
			A2-SF	2994.4840	2994.5185	15146
T.Column	2	26.5313	FbSA-RBF	26.5314	27.0672	1000
			A2-SF	26.5342	26.6429	5884
T.Spring	3	0.0127	FbSA-RBF	0.0127	0.0129	1000
			A2-SF	0.0127	0.0140	7598
T.B.Truss	2	263.8958	FbSA-RBF	263.8959	263.8964	819
			A2-SF	263.9086	264.0120	6649
W.Beam	4	2.3809	FbSA-RBF	2.4568	3.2971	1000
			A2-SF	2.5942	5.2176	11016

The third set of experiments aims to compare the results produced by FbSA-RBF and other methods that use RBF surrogate models, the CARS-RBF, the Constrained Accelerated Random Search algorithm; CPRS-RBF, the RBF-assisted version of the Constrained Pure Random Search and ConstrLMSRBF, an extension of the Local Metric Stochastic RBF reported in [30]. The CARS-RBF and ConstrLMSRBF use surrogates to approximate the objective and constraint functions.

A set of 26 test problems whose dimensions range from 2 to 20 and the number of inequality constraints ranges from 1 to 38 is considered. Fifteen of them belong to the G's collection, a set of well-known constrained optimization test problems developed for the "CEC 2006 Competition on Constrained Real-Parameter Optimization" and described in [26]. The remaining problems are from the suit of scalable functions developed for the "CEC 2010 Competition on Constrained Real-Parameter Optimization", fully described in [29], using the 10-dimensional instances of them. The problems labeled MOD are obtained from the corresponding original version by relaxing the equality constraints. The description of these 26 test problems can be found in [30].

For a fair comparison, the algorithms stop when the maximum number of objective function evaluations $N_{max} = 100(n+1)$ is attained, as suggested in [30]. The results are summarized in Table 3, where f_{avg} and stdv represent, respectively, the mean and the standard deviation of the best objective function values among of the feasible solutions of the 30 runs. In addition, to the FbSA-RBF algorithm, we included the column (succ) with the percentage of runs that produced feasible

solutions (successful runs) among the 30 runs. This information is not available for the other methods in comparison.

FbSA-RB is the algorithm with best f_{avg} results on 15 out of the 26 problems. For example, FbSA-RBF presented much better results than the other algorithms for problems G01, G08, G09, with f_{avg} closer to the global optimal known. It should be noted that for the problem G09, FbSA-RBF has a success rate of 93%, which means that feasible solutions have been obtained in 28 of the 30 runs.

To complement the information of the table, Figure 2 presents the performance profile, as proposed in [13], for the four algorithms for solving this set of 26 problems, related to the performance measure $|f_{avg} - f^*|/max(1, |f^*|)$ as suggested in [11]. The plot reports (on the vertical axis) the percentage of problems solved by each algorithm that is within a certain threshold, t, (on the horizontal axis) of the best result. The higher the percentage the better. A higher value for $t = 1$ means that the corresponding algorithm achieves the smallest value of the metric mostly.

Table 3 Comparison between FbSA-RBF and other methods using RBF surrogate models

		FbSA-RBF				**CARS-RBF**		**CPRS-RBF**		**ConstrLMSRBF**	
Problem	f^*	f_{best}	f_{avg}	stdv	succ	f_{avg}	stdv	f_{avg}	stdv	f_{avg}	stdv
G01	-1.5000e+01	-15.000000	-1.4949e+01	1.0688e-01	100	-8.1982e+00	3.5685e-01	-5.7125e+00	1.4856e-01	-1.2358e+01	3.1299e-01
G02	-8.0362e-01	-0.739492	-5.3021e-01	7.5848e-02	100	-5.4267e-01	1.5301e-02	-4.6623e-01	5.1095e-03	-3.3069e-01	1.7471e-02
G03MOD	-1.0005e+00	-0.995700	-7.8827e-01	3.3974e-01	100	-6.9302e-01	3.7562e-06	-2.1476e-05	1.1090e-05	-6.4612e-01	3.2067e-02
G04	-3.0666e+04	-30665.430000	-3.0514e+04	9.5866e+01	100	-3.0664e+04	1.0583e-01	-3.0512e+04	6.5526e+00	-3.0664e+04	1.0408e-01
G05MOD	5.1265e+03	5125.470000	5.2668e+03	2.1450e+02	47	5.1307e+03	6.8687e-01	5.2860e+03	1.1029e+01	5.1303e+03	5.3444e-01
G06	-6.9618e+03	-6938.087000	-6.8917e+03	2.3192e+01	100	-6.8478e+03	1.0763e+01	-6.1040e+03	7.8520e+01	-6.8843e+03	1.0650e+01
G07	2.4306e+01	24.331230	2.4487e+01	1.0720e-01	100	2.4647e+01	1.9148e-02	5.0061e+02	4.7844e+01	2.4560e+01	1.2624e-02
G08	-9.5825e-02	-0.095825	-9.5822e-02	9.9968e-06	100	-6.2234e-02	6.2338e-03	-8.6539e-02	3.9757e-03	-4.5896e-02	5.5923e-03
G09	6.8063e+02	680.671500	6.8091e+02	1.3249e-01	93	7.7888e+02	1.8201e+01	9.7861e+02	2.3571e+01	9.9629e+02	4.6514e+01
G10	7.0492e+03	7073.403000	7.3638e+03	1.5763e+02	100	7.6393e+03	9.1632e+01	1.3513e+04	2.9508e+02	7.2959e+03	2.9906e+01
G13MOD	5.3942e-02	0.123741	7.4965e-01	2.4279e-01	100	3.5761e-03	1.3628e-05	5.2518e-03	1.2874e-04	1.3880e-02	3.4056e-03
G16	-1.9052e+00	-1.905104	-1.9047e+00	2.6813e-04	100	-1.9032e+00	1.4206e-04	-1.7177e+00	6.3602e-03	-1.9032e+00	1.8738e-04
G18	-8.6603e-01	-0.865987	-8.5837e-01	7.1484e-03	100	-8.1354e-01	1.4906e-02	-2.3243e-01	1.7915e-02	-8.2248e-01	1.3001e-02
G19	3.2656e+01	36.690370	5.4316e+01	1.4838e+01	100	7.7821e+01	2.3259e+00	7.9816e+02	2.3129e+01	6.4197e+01	2.1127e+00
G24	-5.5080e+00	-5.508005	-5.5079e+00	1.3170e-04	100	-5.5061e+00	2.6636e-04	-5.4979e+00	9.8807e-04	-5.3249e+00	8.6699e-02
C01	-7.4731e-01	-0.737633	-5.2942e-01	9.3865e-02	100	-4.5715e-01	1.3609e-02	-4.1669e-01	8.7688e-03	-2.9030e-01	1.0710e-02
C02MOD	-2.2777e+00	-2.191926	-1.2200e+00	5.1038e-01	100	-2.2114e+00	3.1975e-02	-1.0889e+00	3.3414e-02	-2.0195e+00	6.6629e-02
C05MOD	-4.8361e+02	187.537000	4.6127e+02	1.2256e+02	83	-4.7165e+02	4.0030e+00	-2.9240e+02	6.5636e+00	-4.0531e+02	2.1516e+01
C06MOD	-5.7866e+02	379.907600	5.4645e+02	7.7265e+01	93	-3.6983e+02	1.7466e+01	-1.5177e+02	3.7059e+00	-3.2976e+02	2.2255e+01
C07	0.0000e+00	6.2290e+00	1.1605e+02	1.6964e+02	100	3.6011e+05	7.4414e+04	8.1804e+08	1.3970e+08	1.0590e+06	8.9464e+04
C08	0.0000e+00	1.2497e+01	1.6417e+02	1.5569e+02	100	5.7753e+05	1.5980e+05	8.8208e+08	1.4292e+08	2.0133e+06	3.0348e+05
C09MOD	0.0000e+00	4.3717e+12	1.7233e+13	7.0639e+12	100	3.6364e+07	7.0368e+06	6.2069e+10	9.4480e+09	1.7092e+08	1.1831e+07
C10MOD	0.0000e+00	3.0016e+12	1.6665e+13	6.5694e+12	100	3.5231e+07	5.3131e+06	5.1332e+10	8.1646e+09	1.6627e+08	1.3735e+07
C14	0.0000e+00	105.564200	3.5730e+03	4.0722e+03	100	8.8915e+10	7.2741e+10	4.1754e+13	6.1619e+12	3.1018e+09	2.2457e+08
C15	0.0000e+00	263.484300	1.2037e+13	2.3835e+13	100	1.6534e+14	1.4784e+13	1.5982e+14	1.4021e+13	1.6565e+14	1.4788e+13
C17MOD	0.0000e+00	114.687300	5.0187e+02	1.7574e+02	100	4.8892e+00	5.6902e-01	3.6618e+01	2.4068e+00	5.7031e+00	7.7777e-01

So, in terms of efficiency we can say that FbSA-RBF, CARS-RBF and ConstrLMSRBF solved about 58%, 31% and 15% of the problems, respectively, with the smallest value of the metric considered. On the other hand, CPRS-RBF does not solve any problem with the best performance measure. FbSA-RBF and ConstrLMSRBF are the most robust algorithm, solving all problems using no more than 4×10^5 and 8×10^5, respectively, the performance measure used by the best algorithm. Consequently, FbSA-RBF has competitive results when comparing with other strategies using RBF surrogate models.

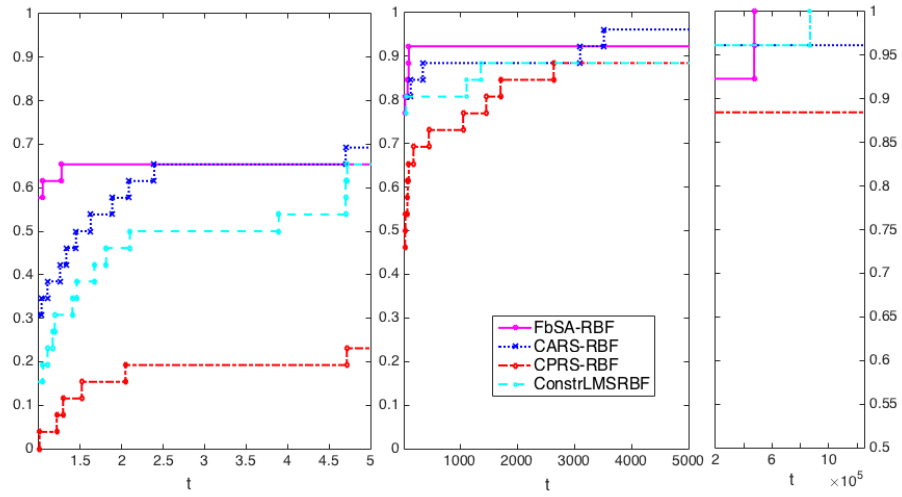

Fig. 2 Performance profile for FbSA-RBF, CARS-RBF, CPRS-RBF and ConstrLMSRBF with measure $|f_{avg} - f^*| \,/\, \max(1, |f^*|)$

5 Conclusions

In this paper, the general FbSA to solve nonlinear and nonconvex constrained global optimization problems is proposed. The convergence of the algorithm is almost surely guaranteed once the probability distribution used in the computation of the iterates satisfies some hypotheses. To optimize problems with computationally expensive black-box objective functions, we developed the FbSA-RBF based on the structure of the general algorithm and assisted by radial basis function surrogate models to approximate the objective function. In each iteration of the FbSA-RBF, several trial points are computed by adding random perturbations with normal distribution in the dynamically chosen coordinates of the best current point. The best of them is chosen by using a non-dominance criterion based on the values of the surrogate model and the infeasibility measure at the trial points. This point replaces the current best point if it is accepted by the filter and does not increase the infeasibility measure by more than a small positive quantity ε_h,

from the current best one. Theoretical results concerning the sufficient conditions for the almost surely convergence of the algorithm are proved. A performance comparison of the FbSA-RBF with other algorithms in the literature, when solving 55 problems of three different sets, is presented. The numerical experiments evidenced significant savings in the number of objective function evaluations by the FbSA-RBF, providing competitive results when compared to other algorithms.

Acknowledgments.

The authors are grateful to the anonymous referees for their fruitful comments and suggestions. The first and second authors were partially supported by Brazilian Funds through CAPES and CNPq by Grants PDSE 99999.009400/2014-01 and 309303/2017-6. The research of the third and fourth authors were partially financed by Portuguese Funds through FCT (Fundação para Ciência e Tecnologia) within the Projects UIDB/00013/2020 and UIDP/00013/2020 of CMAT-UM and UIDB/00319/2020.

References

1. A. H. Aguirre, S. B. Rionda, C. A. Coello Coello, G. L. Lizrraga, and E. M. Montes. Handling constraints using multiobjective optimization concepts. *Int. J. Numer. Meth. Eng.*, 59(15):1989–2017, 2004.
2. M. M. Ali and M. Golalikhani. An electromagnetism-like method for nonlinearly constrained global optimization. *Comput. Math. Appl.*, 60(8):2279 – 2285, 2010.
3. M. M. Ali and W. X. Zhu. A penalty function-based differential evolution algorithm for constrained global optimization. *Comput. Optim. Appl.*, 54(1):707–739, 2013.
4. C. Audet and J. E. Dennis Jr. A pattern search filter method for nonlinear programming without derivatives. *SIAM J. Optimiz.*, 14(4):980–1010, 2004.
5. H. J. C. Barbosa and A. C. C. Lemonge. An adaptive penalty method for genetic algorithms in constrained optimization problems. In H. Iba, editor, *Frontiers in Evolutionary Robotics*, chapter 2. IntechOpen, Rijeka, 2008.
6. E. G. Birgin, C. A. Floudas, and J. M. Martínez. Global minimization using an augmented Lagrangian method with variable lower-level constraints. *Math. Program.*, 125:139–162, 2010.
7. A. J. Booker, J. E. Dennis, P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Struct. Multidisc. Optim.*, 17:1–19, 1999.
8. C. M. Chin and R. Fletcher. On the global convergence of an SLP-filter algorithm that takes EQP steps. *Math. Program.*, 96(1):161–177, 2003.
9. C. A. C. Coello. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Comput. Method. Appl. M.*, 191(11):1245 – 1287, 2002.
10. M. F. P. Costa, F. P. Fernandes, and A. M. A. C. Rocha. Multiple solutions of mixed variable optimization by multistart Hooke and Jeeves filter method. *Appl. Math. Sci.*, 8:2163–2179, 2014.
11. M. F. P. Costa, A. M. A. C. Rocha, and E. M. G. P. Fernandes. Filter-based direct method for constrained global optimization. *J. Global Optim.*, 71(3):517–536, 2018.
12. G. Di Pillo, S. Lucidi, and F. Rinaldi. An approach to constrained global optimization based on exact penalty functions. *J. Global Optim.*, 54:251–260, 2012.
13. E. D. Dolan and J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.*, 91(2):201–213, 2002.
14. N. Echebest, M. L. Shuverdt, and R. P. Vignau. An inexact restoration derivative-free filter method for nonlinear programming. *Comput. Appl. Math.*, 2015. DOI 10.1007/s40314-015-0253-0.
15. P. S. Ferreira, E. W. Karas, M. Sachine, and F. N. Sobral. Global convergence of a derivative-free inexact restoration filter algorithm for nonlinear programming. *Optimization*, 66:271–292, 2017.

16. R. Fletcher, N. I. M. Gould, S. Leyffer, P. L. Toint, and A. Wachter. Global convergence of trust-region SQP-filter algorithm for general nonlinear programming. *SIAM J. Optimiz.*, 13:635–659, 2002.
17. R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Math. Program.*, 91:239–269, 2002.
18. J. Gablonsky. DIRECT version 2.0 user guide. Technical Report CRSC-TR01-08, Center for Research in Scientific Computation, North Carolina State University, 2001.
19. M. L. N. Gonçalves, J. G. Melo, and L. F. Prudente. Augmented Lagrangian methods for nonlinear programming with possible infeasibility. *J. Global Optim.*, 63:297–318, 2015.
20. C. C. Gonzaga, E. W. Karas, and M. Vanti. A globally convergent filter method for nonlinear programming. *SIAM J. Optimiz.*, 14(3):646–669, 2003.
21. N. I. M. Gould, S. Leyffer, and P. L. Toint. A multidimensional filter algorithm for nonlinear equations and nonlinear least-squares. *SIAM J. Optimiz.*, 15(1):17–38, 2004.
22. G. W. Greenwood and Q. J. Shu. Convergence in evolutionary programs with self-adaptation. *Evol. Comput.*, 9(2):147–157, 2001.
23. Q. He and L. Wang. A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Appl. Math. Comput.*, 186(2):1407 – 1422, 2007.
24. A.-R. Hedar and M. Fukushima. Derivative-free filter simulated annealing method for constrained continuous global optimization. *J. Glob. Optim.*, 35(4):521–549, 2006.
25. E. W. Karas, A. P. Oening, and A. A. Ribeiro. Global convergence of slanting filter methods for nonlinear programming. *Appl. Math. Comput.*, 200:486–500, 2008.
26. J. J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. N. Suganthan, C. A. C. Coello, and K. Deb. Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization. Technical report, Nanyang Technological University, Singapore, 2006.
27. J. Long and S. Zeng. A new Filter-Levenberg-Marquart method with disturbance for solving nonlinear complementarity problems. *Appl. Math. Comput.*, 216(2):677–688, 2010.
28. M. J. F. G. Macêdo, M. F. P. Costa, A. M. A. C. Rocha, and E. W. Karas. Combining filter method and dynamically dimensioned search for constrained global optimization. In O. Gervasi, B. Murgante, S. Misra, G. Borruso, C. Torre., A. Rocha, D. Taniar, B. Apduhan, E. Stankova, and A. Cuzzocrea, editors, *Computational Science and Its Applications – ICCSA 2017: 17th International Conference, Trieste, Italy, July 3-6, 2017, Proceedings, Part III*, pages 119–134. Springer International Publishing, 2017.
29. R. Mallipeddi and P. N. Suganthan. Problem definitions and evaluation criteria for the CEC 2010 competition on constrained real-parameter optimization. Technical report, Nanyang Technological University, Singapore, 2010.
30. L. Nuñez, R. G. Regis, and K. Varela. Accelerated random search for constrained global optimization assisted by radial basis function surrogates. *J. Comput. Appl. Math.*, 340:276–295, 2018.
31. G. A. Periçaro, A. A. Ribeiro, and E. W. Karas. Global convergence of a general filter algorithm based on an efficiency condition of the step. *Appl. Math. Comput.*, 219:9581–9597, 2013.
32. Y. G. Petalas, K. E. Parsopoulos, and M. N. Vrahatis. Memetic particle swarm optimization. *Ann. Oper. Res.*, 156(1):99–127, 2007.
33. M. J. D. Powell. The theory of radial basis function approximation in 1990. In W. Light, editor, *Advances in numerical analysis. Vol. 2.Wavelets, subdivision algorithms and radial basis functions*, pages 105–210. Oxford University Press, 1992.
34. C. J. Price, , M. Reale, and B. L. Robertson. Stochastic filter methods for generally constrained global optimization. *J. Global Optim.*, 65(3):441–456, 2016.
35. R. G. Regis. Convergence guarantees for generalized adaptive stochastic search methods for continuous global optimization. *Eur. J. Oper. Res.*, 207:1187–1202, 2010.
36. R. G. Regis. Stochastic radial basis function algorithms for large-scale optimization involving expensive black-box objective and constraint functions. *Comput. Oper. Res.*, 38:837–853, 2011.
37. R. G. Regis. Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. *Eng. Optimiz.*, 46:218–243, 2014.
38. R. G. Regis and C. A. Shoemaker. A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS J. Comput.*, 19(4):497–509, 2007.
39. R. G. Regis and C. A. Shoemaker. Combining radial basis function surrogates and dynamic coordinate search in high-dimensional expensive black-box optimization. *Eng. Optimiz.*, 45:529–555, 2013.

40. S. I. Resnick. *A Probablity Path.* Birkhauser, Boston, 1999.
41. A. A. Ribeiro, E. W. Karas, and C. C. Gonzaga. Global convergence of filter methods for nonlinear programming. *SIAM J. Optimiz.*, 19(3):1231–1249, 2008.
42. A. M. A. C. Rocha, M. F. P. Costa, and E. M. G. P. Fernandes. A filter-based artificial fish swarm algorithm for constrained global optimization: theoretical and practical issues. *J. Global Optim.*, 60:239–263, 2014.
43. A. M. A. C. Rocha and E. M. G. P. Fernandes. Hybridizing the electromagnetism-like algorithm with descent search for solving engineering design problems,. *Int. J. Comput. Math.*, 86:1932–1946, 2009.
44. B. A. Tolson, M. Asadzadeh, and A. Zecchin. Hybrid discrete dynamically dimensioned search (HD-DDS) algorithm for water distribution system design optimization. *Water Resour. Res.*, 45, 2009.
45. B. A. Tolson and C. A. Shoemaker. Dynamically dimensioned search algorithm for computationally efficient watershed model calibration. *Water Resour. Res.*, 43, 2007.
46. M. Ulbrich, S. Ulbrich, and L. N. Vicente. A globally convergent primal-dual interior-point filter method for nonlinear programming. *Math. Program.*, 100(2):379–410, 2004.
47. C.-Y. Wang and D. Li. Unified theory of augmented lagrangian methods for constrained global optimization. *J. Glob. Optim.*, 44(3):433–458, 2008.
48. X. Wang, Z. Zhu, S. Zuo, and Q. Huang. An SQP-filter method for inequality constrained optimization and its global convergence. *Appl. Math. Comput.*, 217(24):10224–10230, 2011.
49. K. Q. Ye, W. Li, and A. Sudjianto. Algorithmic construction of optimal symmetric latin hypercube designs. *J. Stat. Plan. Infer.*, 90(1):145–159, 2000.