Combining Filter Method and Dynamically Dimensioned Search for Constrained Global Optimization

M. Joseane F. G. Macêdo[1], M. Fernanda P. Costa[2], Ana Maria A.C. Rocha[3], and Elizabeth W. Karas[4]

[1] Department of Exact and Natural Sciences,
Federal Rural University of Semi-Árido, 59625-900, Mossoró, Brazil
joseane@ufersa.edu.br
[2] Centre of Mathematics, Department of Mathematics and Applications,
University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal
mfc@math.uminho.pt
[3] Algoritmi Reasearch Centre, Department of Production and Systems,
University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal
arocha@dps.uminho.pt
[4] Department of Mathematics,
Federal University of Paraná, 81531-980, Curitiba, Brazil
ewkaras@ufpr.br

Abstract. In this work we present an algorithm that combines the filter technique and the dynamically dimensioned search (DDS) for solving nonlinear and nonconvex constrained global optimization problems. The DDS is a stochastic global algorithm for solving bound constrained problems that in each iteration generates a randomly trial point perturbing some coordinates of the current best point. The filter technique controls the progress related to optimality and feasibility defining a forbidden region of points refused by the algorithm. This region can be given by the flat or slanting filter rule. The proposed algorithm does not compute or approximate any derivatives of the objective and constraint functions. Preliminary experiments show that the proposed algorithm gives competitive results when compared with other methods.

Keywords: Global optimization. Dynamically dimensioned search algorithm. Filter methods

1 Introduction

Many engineering optimization problems are complex, nonconvex and nonsmooth. Additionally some of them are defined as black-box problems that involve computational expensive computer simulations for which one cannot accurately approximate derivatives. To solve this type of problems, traditional gradient-based methods might not be suitable. In this context, stochastic methods are attractive because they are computationally simple and derivative-free.

Most stochastic methods were primary developed for unconstrained problems and then extended to constrained ones. An important class of methods for constrained optimization are the penalty methods, which seek the solution by replacing the original constrained problem by a sequence of unconstrained subproblems, where the constraint functions are combined with the objective function to define a penalty function.

An alternative to penalty methods to handle constrained optimization problems is the filter method introduced by Fletcher and Leyffer in [1]. This method is based on the concept of dominance, borrowed from multicriteria optimization, to build a filter that accepts iterates if they improve the objective function or improve a constrained violation function, based on *Pareto domination rule*. One of the advantages of this method when compared to the penalty method is that it avoids the initialization and updating of the penalty parameters that are associated with the penalization of the constraints. Filter methods have been combined with trust-region approaches [2,3], SQP techniques [4,5], inexact restoration algorithms [6,7], interior point strategies [8] and line-search algorithms [9,10,11]. They also have been extended to other areas of optimization such as nonlinear equations and inequalities [12,13,14,15], nonsmooth optimization [16,17], unconstrained optimization [18,19], complementarity problems [20,21] and derivative-free optimization [22,23,24,25].

The Dynamically Dimensioned Search (DDS) algorithm was introduced in [26] for automatic calibration of watershed simulation models (WDS). Since the calibration problems have many parameters/variables, DDS proved to be a simple and robust tool for computationally expensive models. In [27], a new global optimization algorithm for WDS optimization, called hybrid discrete dynamically dimensioned search, combines two local search heuristics with a discrete DDS search strategy adapted from the continuous DDS algorithm. Another derivative-free heuristic based on DDS algorithm for optimization of expensive black-box objective functions subject to inequality constraints using surrogate model-based methods was developed in [28,29].

Our contribution is to extend the DDS Algorithm for nonlinear nonconvex and nonsmooth constrained global optimization problems by incorporating a filter technique to handle the constraints.

The problem to be addressed is of the following form:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g(x) \leq 0 \\ & x \in \Omega, \end{array} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^m$ are nonlinear functions and $\Omega = \{x \in \mathbb{R}^n : -\infty < \ell \leq x \leq u < \infty\}$. Since we do not assume that the functions f and g are differentiable and convex, many local minima may exist in the feasible region $\Omega_F = \{x \in \Omega : g(x) \leq 0\}$.

The paper is organized as follows. Section 2 briefly describes the DDS algorithm and Section 3 presents the filter technique to handle the constrained optimization problem. Section 4 presents the proposed algorithm that combines

the DDS algorithm with the filter technique. In Section 5, numerical experiments for two sets of test problems are reported. The paper is concluded in Section 6.

2 Dynamically Dimensioned Search Algorithm

In this section, we briefly describe the DDS algorithm developed by Tolson and Shoemaker [26] for calibration problems that arise in the context of watershed simulation models. These type of problems involve many parameters to estimate (that correspond to decision variables), and are modeled as bound constrained optimization problems. Thus, the DDS Algorithm aims to solve a bound constrained global optimization problem in the following form

$$\underset{x \in \Omega}{\text{minimize}} \quad f(x),$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a nonlinear function in the compact set Ω.

The DDS algorithm is a point-to-point stochastic based heuristic global search algorithm. The main features of the DDS are no parameter tuning and the search strategy of finding good global solutions is scaled within a user-specified maximum number of function evaluations (k_{max}). At the beginning, the algorithm searches globally, and becomes a more local search as the number of iterations approaches the maximum allowable number of function evaluations. 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. Thus, at each iteration k, a trial candidate solution/point x^k_{trial} is obtained by perturbing the current best x^k_{best} only in the randomly selected dimensions. These perturbation magnitudes are randomly sampled from a normal distribution with a mean of zero and a standard deviation σ. Therefore, having selected a subset $I_{perturb}$ of dimensions to be perturbed, the trial x^k_{trial} is componentwised computed by:

$$x^k_{trial,i} = \begin{cases} x^k_{best,i} + N(0, \sigma_i^2) & \text{if } i \in I_{perturb} \\ \\ x^k_{best,i} & \text{otherwise} \end{cases} \tag{2}$$

with a standard deviation $\sigma_i = r(u_i - \ell_i)$ being r a scalar neighborhood size perturbation parameter. Here $x^k_{trial,i}$, $x^k_{best,i}$ and σ_i denote the ith coordinate of the respective vector. If $x^k_{trial} \notin \Omega$ then the trial point is projected onto Ω.

To choose the new best point for the next iteration a greedy procedure between the current best x^k_{best} and the trial candidate x^k_{trial} is performed. The current best x^k_{best} and the trial x^k_{trial} are compared with each other and if the trial decreases the objective function value ($f(x^k_{trial}) < f(x^k_{best})$), then the trial candidate is moved to the next iteration as the current best point; otherwise, the current best is preserved to the next iteration. A formal description of the DDS algorithm is presented in Algorithm 1.

Algorithm 1 DDS Algorithm

Require: $r = 0.2$, k_{max}

Initialization
  Generate $x^1 \in \Omega$
  Set $k = 1$, $x^k_{best} = x^k$, $f^k_{best} = f(x^k)$, $I_{perturb} = \emptyset$
while $k \leq k_{max}$ **do**
  Compute the probability of perturbing the decision variables:
    $P_k = 1 - \dfrac{\ln(k)}{\ln(k_{max})}$
  Select coordinates to perturb:
    Generate uniform random numbers $\omega_i \in [0, 1]$, for $i = 1, \ldots, n$
    Set $I_{perturb} = \{i : \omega_i < P_k\}$
    if $I_{perturb} = \emptyset$ **then**
      Select a random number $i \in \{1, \cdots, n\}$
      Set $I_{perturb} = \{i\}$
    endif
  Generate x^k_{trial} using (2)
  Project x^k_{trial} onto Ω, if necessary
  Evaluate $f(x^k_{trial})$
  Select the best point:
    if $f(x^k_{trial}) < f^k_{best}$ **then**
      Set $x^{k+1}_{best} = x^k_{trial}$, $f^{k+1}_{best} = f(x^k_{trial})$
    endif
  Set $k = k + 1$
end while

3 Filter Methods

In this section we give a brief description of filter methods originally introduced by Fletcher and Leyffer in [1] as an alternative way for globalizing nonlinear programming methods without using any penalty or merit function. The filter methods avoid the problematic issues related to the initial value of the penalty parameter and the tuning of their values throughout the iterative process, which greatly affects the performance of the algorithms.

Filter methods regard (1) as a bi-objective optimization problem that minimizes both the objective function f and a non-negative constraint violation function $h : \mathbb{R}^n \to \mathbb{R}^+$ defined by

$$h(x) = \|g^+(x)\| \tag{3}$$

where $\|g^+(x)\| = \|\max\{0, g(x)\}\|$ for some norm. The filter methods use the concept of dominance from multiobjective optimization, attempting to minimize both functions, but a certain emphasis is put on h, since convergence to a feasible point must be ensured. According to [1], a point x^j (or the corresponding pair $(f(x^j), h(x^j))$ dominates a point x^ℓ (or the corresponding pair $(f(x^\ell), h(x^\ell))$ if

$$f(x^\ell) \leq f(x^j) \text{ and } h(x^\ell) \leq h(x^j).$$

A filter $\mathcal{F}$ is a set of pairs $\{(f^j, h^j)\}$ such that no pair is dominated by any of the others, where $(f^j, h^j) = (f(x^j), h(x^j))$. As proposed in [1], to avoid the acceptance of filter entries close to the boundary formed by the set of all pairs dominated by the filter, the acceptability of the pair $(f(x), h(x))$ to the filter must satisfy a sufficient reduction condition in the form of an *envelope* over all $(f^j, h^j) \in \mathcal{F}$:

$$f(x) < f^j - \alpha h^j \quad \text{or} \quad h(x) < (1-\alpha)h^j\,, \quad \forall (f^j, h^j) \in \mathcal{F} \tag{4}$$

where $\alpha \in (0, 1)$ is a given constant. A filter based on this acceptance rule is designated by *flat filter*. In [30] a slightly different acceptance condition for defining this filter envelope is proposed:

$$f(x) < f^j - \alpha h(x) \quad \text{or} \quad h(x) < (1-\alpha)h^j\,, \quad \forall (f^j, h^j) \in \mathcal{F}. \tag{5}$$

A filter based on this acceptance rule is called by *slanting filter*. A typical filter entry is illustrated in Fig. 3, where the shaded area shows the region dominated by the entry (f^j, h^j) and the *envelope*, in a flat filter (on the left) and in a slanting filter (on the right).

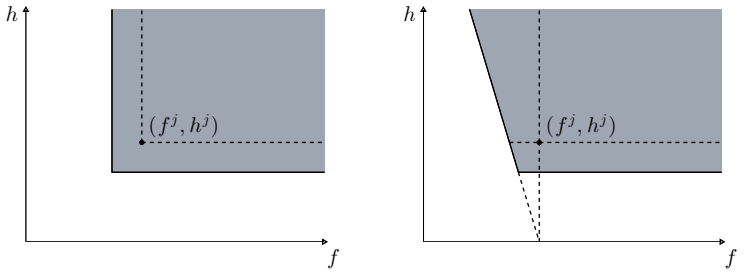

Fig. 1. A typical flat and slanting filter on one entry

Usually, the filter is initialized to be empty $\mathcal{F}_1 = \emptyset$ or with one filter entry that imposes an upper bound for acceptable values on h, $\mathcal{F}_1 = \{(-\infty, h_{\max})\}$ for some $h_{\max} \geq h^1$. In each iteration k, a point x^{k+1} is computed in a such way that the pair (f^{k+1}, h^{k+1}) satisfies the condition (4) (or (5)) for $\mathcal{F} = \mathcal{F}_k \cup \{(f^k, h^k)\}$. If the constraint violation is small and a predicted reduction on f holds, then instead of the filter acceptance criterion, a sufficient reduction on f is required. In case of success, this type of iteration is called a f-type iteration and all the others as h-type iteration (see [1] for more details). On the other hand, in [2,6], an iteration is considered an h-type whenever the function f increases along of the iteration, and all the others as f-type iteration. Such classification of the iterations is used for updating the filter. Throughout the optimization, the filter is updated only at h-type iterations, i.e., in such iterations the pair (f^k, h^k) is added to filter $\mathcal{F}_k$. If the filter is not updated, it remains unchanged, i.e,

$\mathcal{F}_{k+1} = \mathcal{F}_k$ and the entry $\{(f^k, h^k)\}$ is discarded. Thus, the filter is updated as follows:

$$\mathcal{F}_{k+1} = \begin{cases} \mathcal{F}_k \cup \{(f^k, h^k)\} & \text{if } k \text{ is an } h\text{-type iteration} \\ \mathcal{F}_k & \text{otherwise.} \end{cases} \tag{6}$$

Whenever a new entry (f^k, h^k) is added to the filter, all entries that are dominated by (f^k, h^k) are removed from the filter. Furthermore, the updating (6) prevents the addition of feasible iterates to the filter. In fact, whenever x^k is a feasible point then x^{k+1} must decrease f to be accepted by the filter $\mathcal{F}_k \cup \{(f^k, 0)\}$. So, the iteration is a f-type, and consequently the entry $(f^k, 0)$ is discarded.

4 The Proposed Algorithm

In the previous sections we presented the DDS algorithm for bound constrained optimization, and the filter technique for handling constraints. We now present a global algorithm that combines the filter methodology and the DDS, for solving nonlinear constrained problems. The proposed algorithm uses the filter methodology that is able to guarantee sufficient progress towards feasible and optimal solutions of (1), by exploring both feasible and infeasible regions. Furthermore, the filter method allows to select the best non-dominated points, being the least infeasible ones. To promote the exploration of other promising areas of the search region, the algorithm is enriched with a poll-search. Whenever the current best point is a non-dominated feasible point and the algorithm fails in finding a new non-dominated feasible one, then the poll-search is invoked. This procedure searches in a vicinity of the least infeasible point found up to the current iteration k, with the hope to approach a different part of the feasible region. The least infeasible point will be denoted by x_I^k, and the corresponding objective function value and the constraint violation value by f_I^k and h_I^k, respectively. Moreover, at each iteration k, the proposed algorithm uses a set of random trial points $x_{trial_j}^k$ $(j = 1, ..., n_t)$ and uses a new scheme for adjusting the step size.

4.1 Initialization

At iteration $k = 1$, an initial point is randomly generated in the search space Ω as follows

$$x^1 = \ell + \xi(u - \ell), \tag{7}$$

where each component of the vector ξ is a uniformly distributed random number in $[0, 1]$. The initial filter is an empty set, $\mathcal{F}_1 = \emptyset$, the current best point is $x_{best}^1 = x^1$, $(f_{best}^1, h_{best}^1) = (f(x^1), h(x^1))$ and $(f_I^1, h_I^1) = (-\infty, +\infty)$.

4.2 Probability Computation

As in [26], at iteration k, the algorithm computes the probability of perturbing the coordinates of the current best point as follows:

$$P_k = 1 - \frac{\ln(k)}{\ln(k_{max})} \tag{8}$$

where k_{max} is the maximum number of iterations.

4.3 Set of Trial Points

At iteration k, the algorithm computes a set of n_t trial points. First, for each $j \in \{1, \cdots, n_t\}$ are generated n uniform random numbers $\omega_{j,1}, \cdots, \omega_{j,n}$ in $[0, 1]$. These random numbers are used to define the coordinate index set of the current best point x^k_{best} to be perturbed in the following way:

$$I_{perturb_j} = \{i : \omega_{j,i} < P_k\},$$

where the probability P_k is given in (8). If $I_{perturb_j}$ is an empty set, then a coordinate index i is randomly selected from $\{1, ..., n\}$, and $I_{perturb_j} = \{i\}$. Then, the trial point $x^k_{trial_j}$, for $j \in \{1, ..., n_t\}$, is obtained by:

$$x^k_{trial_j,i} = \begin{cases} x^k_{best,i} + \gamma_k N(0, \sigma_i^2) & \text{if } i \in I_{perturb_j} \\ \\ x^k_{best,i} & \text{otherwise,} \end{cases} \tag{9}$$

where $\gamma_k \in (0, 1]$ is a randomization parameter. Note that (9) differs from (2) by this parameter that adjusts the step-size. If $x^k_{trial_j} \notin \Omega$, then the trial point is projected onto Ω. Let $T^k = \{x^k_{trial_1}, ..., x^k_{trial_{n_t}}\}$ be the set of the trial points generated.

4.4 Selection of the Non-Dominated Trial Best Points

Given the set of trial points $T^k = \{x^k_{trial_1}, ..., x^k_{trial_{n_t}}\}$, identify the set of non-dominated trial points $T^k_{ND} \subseteq T^k$, i.e., the points $x^k_{trial_j} \in T^k$ that verify the condition (4) (or (5)) for $\mathcal{F} = \mathcal{F}_k \cup \{(f^k_{best}, h^k_{best})\}$.

If $T^k_{ND} \neq \emptyset$, then two trial best points, denoted by x^k_{bt}, and x^k_{It}, are selected using the following definition:

Definition 1. *(trial best points)*

1. *If there exist feasible points in T^k_{ND}, then x^k_{bt} is the feasible trial point with the smallest objective function value.*
2. *If there exist no feasible points in T^k_{ND}, then x^k_{bt} is the trial point with the least constraint violation value.*
3. *If $T^k_{ND} \backslash \{x^k_{bt}\} \neq \emptyset$, then x^k_{It} is the trial point with the least constraint violation value in this set.*

If $T^k_{ND} = \emptyset$, then the poll-search will be invoked.

4.5 Updating the Best Point

Given x^k_{bt}, the new best point is updated when $h(x^k_{bt}) \leq h(x^k_{best})$, which means that $x^{k+1}_{best} = x^k_{bt}$. If a new best point is obtained, then the iteration k is declared successful. Otherwise, the poll-search will be invoked with x^k_I, the least infeasible point found up to the iteration k. This point is updated as $x^k_I = x^k_{It}$ when $0 < h(x^k_{It}) < h(x^k_I)$.

4.6 Poll-Search

In the poll-search we set $T^k = T^k_{ND} = \emptyset$ and the vicinity of x^k_I is explored. Here, new n_t trial points are randomly generated using (9) but considering x^k_I instead of x^k_{best}. Then, a new set of non-dominated trial points T^k_{ND} is identified. If $T^k_{ND} \neq \emptyset$, new points x^k_{bt} and x^k_I are selected using Definition 1. If the poll-search also fails in finding a new non-dominated point, the iteration k is declared unsuccessful and the step size γ_k will be reduced.

4.7 Updating the Step Size

The parameter γ_k is updated using the following formula:

$$\gamma_{k+1} = \begin{cases} \mu\gamma_k & \text{if } k \text{ is an unsuccessful iteration} \\ \gamma_k & \text{otherwise,} \end{cases} \tag{10}$$

where $0 < \mu < 1$. This parameter controls the randomness, or equivalently, the diversity of the generated points. At the beginning of the iterative process, the parameter must take large values to increase the diversity of the points. Throughout the optimization process its value must decrease in order to fine tuning the points since the effort is focused on exploitation.

4.8 Updating the Filter

Whenever the iteration k is successful, the filter $\mathcal{F}_k$ is updated using formula (6). More precisely, x^{k+1}_{best} verifies the condition (4) (or (5)) for $\mathcal{F}_k \cup \{(f^k_{best}, h^k_{best})\}$ and the filter is updated as follows:

$$\mathcal{F}_{k+1} = \mathcal{F}_k \cup \{(f^k_{best}, h^k_{best})\} \quad \text{if} \quad f^{k+1}_{best} \geq f^k_{best} \quad (h\text{-type iteration}). \tag{11}$$

Otherwise, the filter remains unchanged $\mathcal{F}_{k+1} = \mathcal{F}_k$ and (f^k_{best}, h^k_{best}) is discarded. Whenever a new entry (f^k_{best}, h^k_{best}) is added to the filter, all entries dominated by (f^k_{best}, h^k_{best}) are removed from the filter.

4.9 Stopping Criteria

The stopping criterion of the algorithm is based on the constraint violation and objective function values. Thus, the algorithm stops when the best point x^k_{best}, at iteration k, satisfies

$$f(x^k_{best}) \leq f^* + \varepsilon_f \quad \text{and} \quad h(x^k_{best}) \leq \varepsilon_h \tag{12}$$

where $\varepsilon_f > 0$ is the accuracy error bound on the function value, $\varepsilon_h > 0$ is a small tolerance on the constraint violation and f^* is the best-known solution. For other problems where f^* is not known, the absolute difference between the objective function values of consecutive iterations can be compared with ε_f to decide termination. Besides, if (12) does not hold, the algorithm has an alternative stopping rule based on the maximum number of iterations (k_{max}).

4.10 The Algorithm

A formal description of the proposed algorithm is presented in Algorithm 2.

Algorithm 2

Require: $r = 0.2$, $\sigma_{min} > 0$, $\mu \in (0,1)$, $\gamma_1 \in (0,1)$, $\varepsilon_f > 0$, $\varepsilon_h > 0$, n_t, k_{max}

Set $k = 1$, $\mathcal{F}_1 = \emptyset$, $exist = 0$
Initial point:
    Compute a random initial point x^k by (7). Evaluate $(f^k, h^k) = (f(x^k), h(x^k))$
    Set $x^k_{best} = x^k$, $(f^k_{best}, h^k_{best}) = (f^k, h^k)$, $(f^k_I, h^k_I) = (-\infty, \infty)$
while the stopping criterion is not satisfied **do**
    Set $success = 0$
    Compute the probability P_k by (8)
    Generate n_t trial points using (9): $T^k = \{x^k_{trial_j} \mid j = 1, \ldots, n_t\}$
    Evaluate f and h at the trial points: $(f(x^k_{trial_j}), h(x^k_{trial_j}))$ for $j = 1, \ldots, n_t$
    Select the non-dominated trial points:
        $T^k_{ND} = \{x^k_{trial_j} \in T^k \mid (f(x^k_{trial_j}), h(x^k_{trial_j}))$ satisfies (4) (or (5)) for $\mathcal{F}_k \cup \{(f^k_{best}, h^k_{best})\}\}$
    if $T^k_{ND} \neq \emptyset$ **then**
        Select x^k_{bt} and x^k_{It} in T^k_{ND} by Definition 1
        if $h(x^k_{bt}) \leq h^k_{best}$ **then**
            Set $x^{k+1}_{best} = x^k_{bt}$, $(f^{k+1}_{best}, h^{k+1}_{best}) = (f(x^k_{bt}), h(x^k_{bt}))$, $success = 1$
        end if
        if $h(x^k_{It}) < h^k_I$ **then**
            Set $x^k_I = x^k_{It}$, $exist = 1$
        end if
    end if
    if $(success = 0)$ and $(exist = 1)$ and $(h^k_{best} = 0)$ **then**
        Repeat steps 6-8 with x^k_I instead of x^k_{best}. Repeat steps 9-17
    end if
    if $(success = 1)$ **then**
        Set $\gamma_{k+1} = \gamma_k$
        if $f^{k+1}_{best} > f^k_{best}$ **then**
            $\mathcal{F}_{k+1} = \mathcal{F}_k \cup \{(f^k_{best}, h^k_{best})\}$
        else
            $\mathcal{F}_{k+1} = \mathcal{F}_k$
        end if
    else
        Set $x^{k+1}_{best} = x^k_{best}$, $(f^{k+1}_{best}, h^{k+1}_{best}) = (f^k_{best}, h^k_{best})$, $\gamma_{k+1} = \mu\gamma_k$, $\mathcal{F}_{k+1} = \mathcal{F}_k$
    end if
    Set $(f^{k+1}_I, h^{k+1}_I) = (f^k_I, h^k_I)$
    Set $k = k + 1$
end while

5 Numerical Experiments

In this section we report the numerical experiments to illustrate the practical performance of Algorithm 2 with the flat or the slanting filter. First, a set of 20 small constrained global optimization problems are tested, where the number of variables ranges from 2 to 10 and the number of constraints ranges from 1 to 12, described in [31]. Second, nine well-known engineering design problems

presented in [32] are used, where the number of design variables ranges from 2 to 8 and the number of inequality constraints ranges from 1 to 11. The tests were performed in a notebook ASUSTek Intel Core i7-6700HQ, CPU 2.60GHz, with 16GB RAM, 64-bit, using `Matlab` R2015a.

We considered the Algorithm 2 with the two different filters discussed in Section 3: the flat filter based on the rule (4), denoted as `A2-FF`, and the slanting filter based on the rule (5), denoted as `A2-SF`. The following parameters were fixed: $k_{max} = 300$, the maximum number of iterations; $n_t = 5n$, number of trial points (where n is the dimension of the problem); $\gamma_1 = 1$, the initial randomization parameter related to the step size; and $\mu = 0.8$, the constant used in (10). We adopted the stopping criterion (12) with $\varepsilon_f = 10^{-4}$ and $\varepsilon_h = 10^{-8}$, as considered in [25]. Problems with equality constraints $\psi(x) = 0$ were reformulated into inequalities using $\psi(x) - \delta \leq 0$ and $-\psi(x) - \delta \leq 0$, where $\delta = 10^{-4}$.

To put our approach in perspective, the first set of problems were addressed by two distinct solvers: the Filter-based Artificial Fish Swarm algorithm (`P-BF AFS`) proposed in [25] and the Exact Penalty Global Optimization algorithm (`EPGO`) proposed in [33].

Table 1 lists the results for solving the problems from [31] obtained after 30 independent runs. The first columns display the data of the problem: the identification (**P**); the dimension (n); the known global optimum (f^*). The next columns display for each solver the obtained results 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 of number of function evaluations ($n_{f_{avg}}$) to reach the value f_{best}.

Table 1: Numerical results for the problems described in [31]

P	n	f^*	Solver	f_{best}	f_{med}	h_{best}	h_{med}	$n_{f_{avg}}$
1	5	2.9313e-02	`A2-FF`	0.3051	115.1234	8.25e-04	2.22e-04	8355
			`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	`A2-FF`	-398.300	-84.0701	0.00e00	0.00e00	15476
			`A2-SF`	-395.875	-312.4870	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	`A2-FF`	-386.276	-298.406	0.00e00	0.00e00	14127
			`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	`A2-FF`	-738.748	-703.827	0.00e00	0.00e00	14532
			`A2-SF`	-747.021	-702.915	0.00e00	0.00e00	14630
			`P-BF AFS`	-697.452	-657.349	0.00e00	*	7189

Continued on next page

Table 1 *(Continued from previous page)*

P	n	f^*	Solver	f_{best}	f_{med}	h_{best}	h_{med}	$n_{f_{avg}}$
			EPGO	-82.977		8.43e-04		102015
2d	10	-4.0000e02	A2-FF	-399.234	-381.142	0.00e00	0.00e00	13682
			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	A2-FF	-0.3886	-0.3746	0.00e00	0.00e00	16545
			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	A2-FF	-0.3888	-0.3881	0.00e00	0.00e00	6499
			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	A2-FF	-6.6666	-5.8325	0.00e00	0.00e00	5726
			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	A2-FF	201.159	201.159	1.26e-06	4.43e-02	3074
			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	A2-FF	376.302	376.905	0.00e00	0.00e00	5872
			A2-SF	376.305	376.986	0.00e00	0.00e00	6079
			P-BF AFS	376.293	376.304	0.00e00	*	1335
			EPGO	0.4701		2.05e-05		1523
7	2	-2.8284e00	A2-FF	-2.8283	-2.8219	0.00e00	0.00e00	5114
			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
8	2	-1.1870e02	A2-FF	-118.704	-86.402	0.00e00	0.00e00	5854
			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	A2-FF	-13.4018	-13.3906	0.00e00	0.00e00	8466
			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	A2-FF	0.7418	0.7431	0.00e00	0.00e00	5708
			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

Continued on next page

Table 1 *(Continued from previous page)*

P	n	f^*	Solver	f_{best}	f_{med}	h_{best}	h_{med}	$n_{f_{avg}}$
11	2	-5.0000e-01	A2-FF	-0.5000	-0.4995	0.00e00	0.00e00	5533
			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	A2-FF	-16.7255	-15.3324	0.00e00	8.83e-06	4231
			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	A2-FF	267.923	282.729	0.00e00	4.38e-01	4717
			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	A2-FF	-4.5133	-4.4766	0.00e00	0.00e00	8717
			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	A2-FF	0.0000	0.0000	6.90e-05	1.21e-02	4501
			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	A2-FF	0.7049	0.7050	0.00e00	0.00e00	138
			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

*not available

From the results, we may conclude that the proposed algorithm performs reasonably well. It is able to reach the target solution with good accuracy, except for Problems 1, 2*b* and 13. Therefore, the Algorithm 2 reached the f^* solution in 17 out of the 20 problems, while the P-BF AFS and the EPGO reached the best known solution in 16 and 13 problems, respectively. When we compare our results with those in P-BF AFS, we conclude that the quality of the obtained solutions is comparable although our algorithm required a larger number of function evaluations. On the other hand, EPGO is the most computationally expensive. When comparing the two solvers based on the Algorithm 2, for six problems the solvers obtained the same f_{best}, whereas the A2-FF finds the best solutions for eight problems while A2-SF for six problems. In terms of $n_{f_{avg}}$, there are no significant differences between the solvers.

Finally, the next experiment aims to show the effectiveness of the proposed algorithm when solving more complex and real application problems. Thus, the

second set of test problems comprises nine problems that arise from well-known engineering design problems described in [32]. Table 2 lists the results obtained after 30 independent runs for the developed solvers `A2-FF` and `A2-SF`. All the parameter settings are the same as for the previous experiment.

Table 2. Numerical results for the problems described in [32]

P	n	f^*	**Solver**	f_{best}	f_{med}	h_{best}	h_{med}	$n_{f_{avg}}$
Cylindrical Vessel	5	5868.7650	`A2-FF`	5978.6504	6384.3347	0.00e00	0.00e00	11151
			`A2-SF`	5898.3626	6327.0383	0.00e00	0.00e00	10966
Disc Brake	4	0.1274	`A2-FF`	0.1274	0.1355	0.00e00	0.00e00	8916
			`A2-SF`	0.1274	0.1283	0.00e00	0.00e00	6770
Four Bar Truss	4	1400.0000	`A2-FF`	1400.0000	1400.0000	0.00e00	0.00e00	276
			`A2-SF`	1400.0000	1400.0000	0.00e00	0.00e00	336
Heat Exchanger	8	7049.2480	`A2-FF`	7144.4819	9383.1025	0.00e00	0.00e00	16781
			`A2-SF`	7075.0293	8340.3915	0.00e00	0.00e00	17826
Speed Reducer	7	2994.4991	`A2-FF`	2994.4840	2994.4971	0.00e00	0.00e00	14771
			`A2-SF`	2994.4840	2994.5185	0.00e00	0.00e00	15146
Tubular Column	2	26.5313	`A2-FF`	26.5386	26.6533	0.00e00	0.00e00	5883
			`A2-SF`	26.5342	26.6429	0.00e00	0.00e00	5884
Tension Spring	3	0.0127	`A2-FF`	0.0127	0.0144	0.00e00	0.00e00	6976
			`A2-SF`	0.0127	0.0140	0.00e00	0.00e00	7598
Three Bar Truss	2	263.8958	`A2-FF`	263.9017	263.9764	0.00e00	0.00e00	6514
			`A2-SF`	263.9086	264.0120	0.00e00	0.00e00	6649
Welded Beam	4	2.3809	`A2-FF`	2.5380	4.2294	0.00e00	0.00e00	10966
			`A2-SF`	2.5942	5.2176	0.00e00	0.00e00	11016

From the results for this set of problems, we may conclude that our algorithm is able to reach the target solution with good accuracy, except for Cylindrical Vessel, Heat Exchanger and Welded Beam problems. Furthermore, the performance of the solvers `A2-FF` and `A2-SF` is similar.

6 Conclusions

This paper presents the DDS algorithm combined with the filter method to solve nonlinear and nonconvex constrained global optimization problems. The DDS algorithm was developed for calibration problems that arise in the context of WSM, modeled as bound constrained optimization problems. The proposed algorithm is an extension of the DDS algorithm incorporating a filter method that reformulates the optimization problem as a bi-objective optimization one, aiming to minimize the objective and the constraint violation functions. The reported numerical results show the effectiveness of the proposed algorithm and its competitive practical performance when compared with a penalty framework and with a filter-based stochastic global AFS algorithm from the literature.

Future developments will focus on the decrease of the computational cost and the solution of larger dimensional problems. A study of the convergence of the algorithm will be carried out in the future.

Acknowledgments. The first author thanks a scholarship supported by the International Cooperation Program CAPES/ COFECUB at the University of Minho. The second and third authors thanks the support given by FCT (Fundação para Ciência e Tecnologia, Portugal) in the scope of the projects: UID/MAT/00013/2013 and UID/CEC/00319/2013. The fourth author was partially supported by CNPq-Brazil grants 308957/2014-8 and 401288/2014-5.

References

1. Fletcher, R., Leyffer, S.: Nonlinear programming without a penalty function. Math. Program. **91**, 239–269 (2002)
2. Periçaro, G.A., Ribeiro, A.A., Karas, E.W.: Global convergence of a general filter algorithm based on an efficiency condition of the step. Appl. Math. Comput. **219**, 9581–9597 (2013)
3. Ribeiro, A.A., Karas, E.W., Gonzaga, C.C.: Global convergence of filter methods for nonlinear programming. SIAM J. Optim. **19**(3), 1231–1249 (2008)
4. Fletcher, R., Gould, N.I.M., Leyffer, S., Toint, P.L., Wachter, A.: Global convergence of trust-region SQP-filter algorithm for general nonlinear programming. SIAM J. Optim. **13**, 635–659 (2002)
5. Wang, X., Zhu, Z., Zuo, S., Huang., Q.: An SQP-filter method for inequality constrained optimization and its global convergence. Appl. Math. Comput. **217**(24), 10224–10230 (2011)
6. Gonzaga, C.C., Karas, E.W., Vanti, M.: A globally convergent filter method for nonlinear programming. SIAM J. Optimiz. **14**(3), 646–669 (2003)
7. Karas, E.W., Oening, A.P., Ribeiro, A.A.: Global convergence of slanting filter methods for nonlinear programming. Appl. Math. Comput. **200**, 486–500 (2008)
8. Ulbrich, M., Ulbrich, S., Vicente, L.N.: A globally convergent primal-dual interior-point filter method for nonlinear programming. Math. Program. **100**(2), 379–410 (2004)
9. Gu, C., Zhu, D.: A secant algorithm with line search filter method for nonlinear optimization. Appl. Math. Model. **35**(2), 879–894 (2011)
10. Pei, Y., Zhu, D.: A trust-region algorithm combining line search filter technique for nonlinear constrained optimization. Int. J. Comput. Math. **91**(8), 1817–1839 (2014)
11. Wächter, A., Biegler, L.T.: Line search filter methods for nonlinear programming: motivation and global convergence. SIAM J. Optimiz. **16**(1), 1–31 (2005)
12. Echebest, N., Shuverdt, M.L., Vignau, R.P.: A derivative-free method for solving box-constrained underdetermined nonlinear systems of equations. Appl. Math. Comput. **219**, 3198–3208 (2012)
13. Fletcher, R., Leyffer, S.: Filter-type algorithms for solving systems of algebraic equations and inequalities. In di Pillo, G., Murli, A., eds.: Advances in Optimization and Numerical Analysis. High Performance Algorithms and Software for Nonlinear Optimization, Kluwer,259–278 (2003)

14. Gould, N.I.M., Leyffer, S., Toint., P.L.: A multidimensional filter algorithm for nonlinear equations and nonlinear least-squares. SIAM J. Optimiz. **15**(1), 17–38 (2004)
15. Gould, N.I.M., Toint., P.L.: FILTRANE, a Fortran 95 filter-trust-region package for solving least-squares and nonlinear feasibility problems. ACM T. Math. Software **33**(1), 3–25 (2007)
16. Karas, E.W., Ribeiro, A.A., Sagastizábal, C., Solodov, M.: A bundle-filter method for nonsmooth convex constrained optimization. Math. Program. **116**, 297–320 (2009)
17. Peng, Y., Feng, H., Li, Q.: A filter-variable-metric method for nonsmooth convex constrained optimization. Appl. Math. Comput. **208**(1), 119–128 (2009)
18. Gould, N.I.M., Sainvitu, C., Toint., P.L.: A filter-trust-region method for unconstrained optimization. SIAM J. Optimiz. **16**(2), 341–357 (2006)
19. Yang, Z., Sun, W.: A filter-trust-region method for LC^1 unconstrained optimization and its global convergence. Anal. Theor. Appl. **24**(1), 55–66 (2008)
20. Long, J., Ma, C., P.Nie: A new filter method for solving nonlinear complementarity problems. Appl. Math. Comput. **185**(1), 705–718 (2007)
21. Long, J., Zeng, S.: A new Filter-Levenberg-Marquart method with disturbance for solving nonlinear complementarity problems. Appl. Math. Comput. **216**(2), 677–688 (2010)
22. Audet, C., Dennis Jr., J.E.: A pattern search filter method for nonlinear programming without derivatives. SIAM J. Optim. **14**(4), 980–1010 (2004)
23. Echebest, N., Shuverdt, M.L., Vignau, R.P.: An inexact restoration derivative-free filter method for nonlinear programming. Comput. Appl. Math. **36**(1), 693–718 (2017)
24. Ferreira, P.S., Karas, E.W., Sachine, M., Sobral, F.N.C.: Global convergence of a derivative-free inexact restoration filter algorithm for nonlinear programming. Optimization **66**, 271–292 (2017)
25. Rocha, A.M.A.C., Costa, M.F.P., Fernandes, E.M.G.P.: A filter-based artificial fish swarm algorithm for constrained global optimization: theoretical and practical issues. J. Global Optim. **60**, 239–263 (2014)
26. Tolson, B.A., Shoemaker, C.A.: Dynamically dimensioned search algorithm for computationally efficient watershed model calibration. Water Resour. Res. **43** (2007)
27. Tolson, B.A., Asadzadeh, M., Zecchin, A.: Hybrid discrete dynamically dimensioned search (hd-dds) algorithm for water distribution system design optimization. Water Resour. Res. **45** (2009)
28. Regis, R.G.: Stochastic radial basis function algorithms for large-scale optimization involving expensive black-box objective and constraint functions. Comput Oper Res **38**, 837–853 (2011)
29. Regis, R.G.: Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. Eng Optimiz **46**, 218–243 (2014)
30. Chin, C.M., Fletcher, R.: On the global convergence of an SLP-filter algorithm that takes EQP steps. Math. Program. **96**(1),161–177 (2003)
31. Birgin, E.G., Floudas, C.A., Martínez, J.M.: Global minimization using an augmented Lagrangian method with variable lower-level constraints. Math. Program. **125**(1), 139–162 (2010)
32. Rocha, A.M.A.C., Fernandes, E.M.G.P.: Hybridizing the electromagnetism-like algorithm with descent search for solving engineering design problems. Int. J. Comput. Math. **86**, 1932–1946 (2009)

33. Pillo, G.D., Lucidi, S., Rinaldi, F.: An approach to constrained global optimization based on exact penalty functions. J. Glob. Optim. **54**, 251–260 (2012)