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

A filter-based artificial fish swarm algorithm for constrained global optimization: theoretical and practical issues

Ana Maria A.C. Rocha · **M. Fernanda P. Costa** ·
Edite M.G.P. Fernandes

Received: date / Accepted: date

Abstract This paper presents a filter-based artificial fish swarm algorithm for solving nonconvex constrained global optimization problems. Convergence to an ε-global minimizer is guaranteed. At each iteration k, the algorithm requires a $(\rho^{(k)}, \varepsilon^{(k)})$-global minimizer of a bound constrained bi-objective subproblem, where as $k \to \infty$, $\rho^{(k)} \to 0$ gives the constraint violation tolerance and $\varepsilon^{(k)} \to \varepsilon$ is the error bound defining the accuracy required for the solution. The subproblems are solved by a population-based heuristic known as artificial fish swarm algorithm. Each subproblem relies on the approximate solution of the previous one, randomly generated new points to explore the search space for a global solution, and the filter methodology to accept non-dominated trial points. Convergence to a $(\rho^{(k)}, \varepsilon^{(k)})$-global minimizer with probability one is guaranteed by probability theory. Preliminary numerical experiments show that the algorithm is very competitive when compared with known deterministic and stochastic methods.

Keywords Global optimization · Artificial fish swarm · Filter method · Stochastic convergence

Ana Maria A.C. Rocha
Department of Production and Systems,
Algoritmi Research Centre,
University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal
E-mail: arocha@dps.uminho.pt

M. Fernanda P. Costa
Department of Mathematics and Applications,
Centre of Mathematics,
University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal
E-mail: mfc@math.uminho.pt

Edite M.G.P. Fernandes
Algoritmi Research Centre
E-mail: emgpf@dps.uminho.pt

1 Introduction

This paper aims to contribute to the research area of constrained global optimization (CGO) by proposing the integration of a filter methodology, as outlined in [21], into a population-based stochastic method to globally solve nonlinear constrained optimization problems. Our main goal is to guarantee that a global optimal solution of a nonconvex constrained optimization problem is obtained, up to a required accuracy $\varepsilon > 0$. The mathematical formulation of the problem is:

$$\begin{array}{rl} \min\limits_{x \in \Omega} & f(x) \\ \text{subject to} & h(x) = 0 \\ & g(x) \leq 0 \end{array} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $h : \mathbb{R}^n \to \mathbb{R}^q$ and $g : \mathbb{R}^n \to \mathbb{R}^p$ are nonlinear continuous functions and $\Omega = \{x \in \mathbb{R}^n : -\infty < l_i \leq x_i \leq u_i < \infty, i = 1, \ldots, n\}$. The feasible region defined by the constraints is denoted by $\Lambda \subseteq \Omega$.

We do not assume that the objective function f and the constraints are convex and thus many local minima may exist in the feasible region. For this class of global optimization problems, methods based on penalty functions are common in the literature [5,7,14,19,28, 31,39,46,55]. In this type of methods, the constraint violation is combined with the objective function to define a penalty function. This function aims at penalizing infeasible solutions by increasing their fitness values proportionally to their level of constraint violation. Penalty functions require the use of a positive penalty parameter that aims to balance function and constraint violation values. However, setting the initial value for the penalty parameter and tuning its values throughout the iterative process are problematic issues since the performance of the algorithm is greatly affected by their values. In some cases, the optimal solution of the problem is attained only when the penalty parameter approaches infinity. Augmented Lagrangian functions are penalty functions for which a finite penalty parameter value is in general sufficient to guarantee convergence to the solution of the constrained problem [8]. An augmented Lagrangian technique has also been extended to global optimization where a hybrid genetic algorithm has been implemented to globally solve the subproblems [15]. Convergence to a global minimizer has been proved. Other augmented Lagrangians for CGO problems have been implemented and their convergence properties have been derived. For example, in [10] a global optimization method based on the augmented Lagrangian function of Powell-Hestenes-Rockafellar (PHR), where the exact α branch-and-bound (BB) method is used to find the approximate global solutions to the subproblems. The usage of the PHR function has been extended to stochastic population-based methods like the electromagnetism-like mechanism of optimization [48] and the artificial fish swarm algorithm [51]. Recently, in [11], global convergence to a minimum is guaranteed using the PHR Lagrangian with a nonmonotone penalty parameter tuning and gradient-based approaches for solving the box constrained subproblems. Other proposals concerning augmented Lagrangian functions for global optimization can be found in [17,26,32,40,53,66].

Other strategies aiming to handle the constraints of the problem are available in the literature. Techniques based on multi-objective concepts, where both constraint violation and objective function are goals to be minimized separately when progressing towards an optimal solution are presented in [1,3]. A gradient-based approach to repair feasibility is proposed in [12]. Arbitrary bounding operations are used to formulate a geometric BB for CGO problems [52]. The use of the filter method to guarantee sufficient progress towards feasible and optimal solutions of nonlinear constrained optimization has its origin in [21]. A variety of filter-based algorithms have been proposed ever since for nonlinear programming [22, 35,47], in direct search methods [6,18], bundle methods [36], interior point barrier methods [59], primal-dual interior point methods [56], in reduction methods for semi-infinite programming [45], in methods for nonlinear complementarity problems [43], constrained mixed variable programming [29], systems of nonlinear equations [23,34,44,60], nonlinear semidefinite programming [25], and to define criteria of iterate acceptance in unconstrained problems [62]. Even nonmonotone versions of filter methods have been implemented with success in the nonlinear programming area [16,24,54,63,65]. The use of filter methods has been indeed spread to all areas of computational mathematics.

Previous work applying the filter set methodology to handle the constraints within stochastic algorithms shows that the idea is promising when compared with other methods existing in the literature [29,30,49]. This paper aims to explore further this research area including a theoretical convergence analysis of a stochastic population-based filter algorithm.

Different types of methods have been proposed to solve global optimization problems. A list of interesting and efficient stochastic algorithms includes the artificial fish swarm (AFS), artificial bee colony (ABC), differential evolution (DE), harmony search, genetic algorithm (GA), evolution strategy, particle swarm optimization (PSO), Cuckoo-search (CK), simulated annealing [2,4,41,57,58,61,64]. A recent conceptual comparison of CK, PSO, DE and ABC algorithms is presented in [13]. The AFS algorithm is a recent and easy to implement artificial life computing algorithm that simulates fish swarm behavior inside water. The algorithm uses a population of points to identify promising regions looking for a global solution [33,50,61]. Recently, a state of art concerned with hybridizations and applications of the AFS algorithm appeared in [42].

In this paper, we aim to integrate the filter methodology into the AFS algorithm for solving CGO problems. This study comes in the sequence of a previous preliminary practical study concerning the AFS and filter methods [49]. In the herein presented algorithm, convergence to an ε-global minimizer is guaranteed. Further, the population-based filter algorithm is conceptually different since a sequence of subproblems with increasing accuracy is required to be solved. At each iteration k, a $(\rho^{(k)},\varepsilon^{(k)})$-global minimizer of a bound constrained bi-objective subproblem, where $\rho^{(k)} \to 0$ gives the constraint violation tolerance and $\varepsilon^{(k)} \to \varepsilon$ is the error bound defining the accuracy required for the solution, as $k \to \infty$, is required. Each subproblem is solved by a filter-based AFS algorithm with guaranteed convergence to a $(\rho^{(k)},\varepsilon^{(k)})$-global minimizer, with probability one.

Another previous study concerned with the convergence of the AFS algorithm shows that convergence in mean square to a certain position in the search space has been guaranteed in [51]. Convergence here means stochastic convergence that relies on a probability measure and is different from the convergence concept of classical analysis. Convergence of random variables can be defined in various ways [37]. For example, convergence with probability one (or almost sure convergence), convergence in probability, convergence in mean square, and convergence in distribution. Each one is qualitatively different. Almost sure convergence is the probabilistic version of point-wise convergence known from elementary real analysis. In this study we are particularly interested in convergence with probability one.

To the best of our knowledge, this is the first attempt to analyze the performance and the convergence properties of a population-based stochastic method using a filter methodology to handle the equality and inequality constraints, aiming to globally solve an optimization problem.

The paper is organized as follows. In Section 2, the population-based filter paradigm is presented and its convergence properties are derived. Section 3 describes the filter-based AFS algorithm and discusses its asymptotic convergence properties. Section 4 shows some numerical results and we conclude the paper in Section 5.

2 Population-based filter paradigm

We aim to analyze the behavior of a stochastic global optimization method based on the filter methodology to guarantee convergence to a feasible and optimal global solution of problem (1). Based on a population of points, the herein presented stochastic method generates, at each iteration, an approximate global minimizer. A list of notation follows.

Notation 1 $\|\cdot\|$ *represents the Euclidean norm, the component i of a vector x is represented by* $(x)_i$ *and the component i of the vector* $v_+ \in \mathbb{R}^p$ *is defined by* $\max\{0,(v)_i\}$ *where* $v = ((v)_1,\ldots,(v)_p)^T$. *We use* $\rho^{(k)} \downarrow 0$ *to indicate that the sequence* $\{\rho^{(k)}\}$ *of non-negative decreasing numbers tends to zero. The position of a point in the space is represented by* $x_j \in \mathbb{R}^n$ *(the jth point of a population) and m is the number of points in the population. Let* $X \equiv [x_1 x_2 \cdots x_m] \in \mathbb{R}^{n\times m}$ *be the matrix that contains the m points of the population, where* $m < \infty$. *Let LB denote the smallest function value considering all algorithms that found a feasible solution.*

Given an approximation to the solution of the CGO problem (1), $x^{(k)}$, at iteration k, equality and inequality infeasibility is measured by the non-negative constraint violation function $\theta : \mathbb{R}^n \to \mathbb{R}_+$

$$\theta(x^{(k)}) = \|h(x^{(k)})\|^2 + \|g(x^{(k)})_+\|^2. \tag{2}$$

Definition 1 (ε-global solution to the CGO problem) When finding the global minimum of the continuous optimization problem (1), where $\Omega \subset \mathbb{R}^n$ is a bounded space, the point $\bar{x} \in \Omega$ is an ε-global minimizer of problem (1) if

$$\theta(\bar{x}) = 0 \text{ and } f(\bar{x}) \le \min_{z\in\Lambda\subseteq\Omega} f(z) + \varepsilon, \tag{3}$$

where $\varepsilon > 0$ is the error bound which reflects the accuracy required for the solution.

The proposed population-based filter paradigm uses the filter set concept [21] that is able to explore both feasible and infeasible regions. This technique incorporates the concept of non-dominance, present in the field of multi-objective optimization, to build a filter that is able to accept a trial approximation if it improves either the objective function or the constraint violation, relative to the current approximation. Algorithms based on the filter methodology treat the optimization problem as a bi-objective problem aiming to minimize both the objective function $f(x)$ and the constraint violation function $\theta(x)$.

The herein presented paradigm randomly generates a population of solutions/points and computes an approximate global solution, $x^{(k)}$, denoted by $(\rho^{(k)}, \varepsilon^{(k)})$-global minimizer, to the bound constrained bi-objective optimization subproblem

$$\min_{x\in\Omega} \; (\theta(x), f(x)), \tag{4}$$

then randomly generates another population of points, and continues to solve the next subproblem always using the approximate solution of the previous one. Thus, a sequence of bi-objective optimization subproblems with increasingly better accuracy is solved. According to the definition of the subproblem (4), at each iteration k, where k denotes the iteration counter of this outer cycle, the method to compute a $(\rho^{(k)}, \varepsilon^{(k)})$-global minimizer will ensure that the bound constraints are always satisfied and an approximate global minimum is obtained. Each time a new subproblem (4) is solved, a set of $m-1$ points is randomly generated in the search space, since the approximate solution of the previous subproblem is retained. This is equivalent to the resetting of the population, a well-known strategy aiming to explore the search space for promising regions.

Definition 2 (Approximate global minimizer) For $\rho^{(k)}, \varepsilon^{(k)} > 0$, an iterate $x^{(k)}$ is said to be a $(\rho^{(k)}, \varepsilon^{(k)})$-global minimizer of subproblem (4) if the two following conditions hold:

$$\begin{array}{l} \theta(x^{(k)}) \le \rho^{(k)} \\ f(x^{(k)}) \le f(x) + \varepsilon^{(k)}, \text{ for all } x \in \Lambda \subseteq \Omega. \end{array} \tag{5}$$

When the optimization problem is nonconvex, a global optimization method is required to solve the subproblem (4), so that the algorithm has some guarantee to converge to a global solution instead of being trapped in a local one.

Our proposal for computing a $(\rho^{(k)}, \varepsilon^{(k)})$-global minimizer of subproblem (4), at iteration k, is based on a stochastic population-based algorithm, known as AFS algorithm, and on the filter method that provides selection criteria aiming to accept a sequence of non-dominated trial points. The non-dominated best points of the population will generate the sequence of approximations to the required solution $x^{(k)}$. The algorithm consists of two nested cycles. Each iteration of the outer cycle provides a $(\rho^{(k)}, \varepsilon^{(k)})$-global minimizer of subproblem (4) and updates the error tolerances $\rho^{(k)}$ and $\varepsilon^{(k)}$. Each iteration of the inner cycle generates a population of trial points, checks their acceptability to the filter and selects a non-dominated best point. Iterations of the inner cycle are performed until an acceptable best point satisfies conditions (5).

2.1 Population-based filter algorithm

A formal description of the population-based filter (P-BF) algorithm for solving the original problem (1), based on a filter-based AFS algorithm to solve the sequence of subproblems (4), is presented in Algorithm 1.

Algorithm 1 (P-BF algorithm)

```
Require: $0 < \gamma_\rho < 1$, $0 < \gamma_\varepsilon < 1$, $0 < \rho_{tol} \ll 1$, $0 < \varepsilon \ll 1$, $\rho^{(1)} > \rho_{tol}$, $\varepsilon^{(1)} > \varepsilon$, $LB$;
set $k = 1$;
randomly generate a point $x^{(0)}$ in $\Omega$;
while $\theta(x^{(k-1)}) > \rho_{tol} \ \vee \ f(x^{(k-1)}) > LB + \varepsilon$ do
    find a $(\rho^{(k)}, \varepsilon^{(k)})$-global minimizer $x^{(k)}$ of subproblem (4) using the filter-based AFS algorithm (Algorithm 5) so that
        $\theta(x^{(k)}) \le \rho^{(k)}$
        $f(x^{(k)}) \le f(x) + \varepsilon^{(k)}$, for all $x \in \Lambda \subseteq \Omega$;
    update $\rho^{(k+1)} = \max\left\{\rho_{tol}, \gamma_\rho \rho^{(k)}\right\}$;
    update $\varepsilon^{(k+1)} = \max\left\{\varepsilon, \gamma_\varepsilon \varepsilon^{(k)}\right\}$;
    set $k = k + 1$;
end while
```

The stopping criterion of the algorithm relies on both the constraint violation and objective function values. Thus, the algorithm is stopped when an approximate solution $x^{(k)}$ satisfying

$$\theta(x^{(k)}) \le \rho_{tol} \text{ and } f(x^{(k)}) \le LB + \varepsilon$$

is found, for a sufficiently small tolerance $\rho_{tol} > 0$ and an accuracy error bound on the function value $\varepsilon > 0$. This criterion is applicable when solving test problems. For other problems where the best-known solution LB is not used, the absolute difference between the objective function values of two consecutive iterations can be compared with ε to decide termination. At each iteration k, an approximate solution to the subproblem (4) is required to satisfy the conditions (5), before the algorithm continues finding the solution of the next subproblem.

To be able to converge to an optimal solution of problem (1), the algorithm requires that $\{\rho^{(k)}\}$ defines a monotone decreasing sequence of positive values converging to zero (see Step 5 in Algorithm 1), and the decreasing sequence of positive values $\{\varepsilon^{(k)}\}$ converges to ε, the user required accuracy for the solution, as $k \to \infty$ (see Step 6 in Algorithm 1).

2.2 Convergence to an ε-global minimizer

Here we aim to prove that every limit point of a sequence of iterates generated by the P-BF algorithm is feasible and an ε-global minimizer of problem (1). This convergence analysis follows the reasoning used in the convergence of the global augmented Lagrangian method [10]. We now state the assumptions that are needed to show convergence of the P-BF algorithm (Algorithm 1) to an ε-global minimum. Let $\{x^{(k)}\}$ be the sequence generated by the algorithm.

Assumption A 1 *A global minimizer z of the problem* (1) *exists.*

Assumption A 2 *The sequence $\{x^{(k)}\}$ generated by the Algorithm 1 is well defined and there exists a set of indices $\mathscr{N} \subseteq \mathbb{N}$ so that* $\lim_{k\in\mathscr{N}} x^{(k)} = x^*$.

Assumption A 3 *The functions $f : \mathbb{R}^n \to \mathbb{R}$, $h : \mathbb{R}^n \to \mathbb{R}^q$ and $g : \mathbb{R}^n \to \mathbb{R}^p$ are continuous in Ω.*

Since the set Ω is compact and $\theta(x)$ and $f(x)$ are continuous, a $(\rho^{(k)}, \varepsilon^{(k)})$-global minimizer of subproblem (4), $x^{(k)}$, necessarily exists.

Assumption A 4 *$\{\rho^{(k)}\}$ is a decreasing sequence of non-negative real numbers that tends to zero.*

Assumption A 5 *$\{\varepsilon^{(k)}\}$ is a decreasing sequence of non-negative real numbers that tends to ε.*

We now investigate the properties of the limit points of the sequence $\{x^{(k)}\}$.

Theorem 1 *Assume that the Assumptions A 1 through A 4 hold. Then every limit point x^* of the sequence $\{x^{(k)}\}$ generated by the Algorithm 1 is feasible.*

Proof Since $x^{(k)} \in \Omega$ and Ω is closed then $x^* \in \Omega$. Since $\rho^{(k)} \downarrow 0$ then $\lim_k \theta(x^{(k)}) = 0$. This implies that $\|h(x^{(k)})\| \to 0$ and $\|g(x^{(k)})_+\| \to 0$. So, $h_j(x^{(k)}) \to 0$ for all $j = 1, \ldots, q$ and $g_i(x^{(k)})_+ \to 0$ for all $i = 1, \ldots, p$, and we conclude that the limit point is feasible. □

We now prove that a sequence of iterates generated by the algorithm converges to an ε-global minimizer of the problem (1).

Theorem 2 *Assume that the Assumptions A 1 through A 5 hold. Then every limit point x^* of a sequence $\{x^{(k)}\}$ generated by Algorithm 1 is an ε-global minimizer of the problem* (1).

Proof Let $\mathscr{N} \subset \mathbb{N}$ be the set of indices such that $\lim_{k\in\mathscr{N}} x^{(k)} = x^*$. By Theorem 1, x^* is feasible which means that $\lim_{k\in\mathscr{N}} \theta(x^{(k)}) = \theta(x^*) = 0$. Let $z \in \Omega$ be a global minimizer of problem (1). By the definition of $x^{(k)}$ in the Algorithm 1 we have:

$$f(x^{(k)}) \leq f(z) + \varepsilon^{(k)}.$$

Taking limits for $k \in \mathscr{N}$ and using $\lim_{k\in\mathscr{N}} \varepsilon^{(k)} = \varepsilon$, we obtain:

$$\lim_{k\in\mathscr{N}} f(x^{(k)}) = f(x^*) \leq f(z) + \varepsilon$$

which proves the claim that x^* is an ε-global minimizer, since z is a global minimizer. □

3 Filter-based AFS algorithm for solving the subproblems

Our proposal for solving the subproblem (4) is the AFS algorithm [50,51]. This is a stochastic population-based global optimization algorithm for nonlinear optimization. It relies on a swarm intelligence based paradigm to construct fish/point movements over the search space. The purpose is to find an approximate global solution $x^{(k)}$ of subproblem (4) satisfying (5). To minimize θ and f, a filter method is used. This is an efficient methodology that is applied to nonlinear constrained optimization (NCO) problems aiming to build a region of prohibited points when minimizing infeasibility and objective function values [6,18,21,59]. In a filter method, both θ and f are required to be reduced. This technique has been used as an alternative to merit functions to enforce progress towards the solution of NCO problems. The underlying concept is that trial points are accepted as possible solutions if they improve the constraint violation or the objective function value.

A filter $\mathscr{F}$ is a finite set of points x_j, corresponding to pairs $(\theta(x_j), f(x_j))$, none of which is dominated by any of the others. The concept of dominance arises from the multi-objective optimization area. According to [21], the used definition is the following.

Definition 3 (Dominance) A point x, or the corresponding pair $(\theta(x), f(x))$, is said to dominate y, or the corresponding pair $(\theta(y), f(y))$, denoted by $x \prec y$, if and only if

$$\theta(x) \leq \theta(y) \text{ and } f(x) \leq f(y).$$

In the context of the filter set, a point y is said to be filtered if $x \prec y$ for some $x \in \mathscr{F}$, or if $\theta(y) \geq \theta_{\max}$, for some positive upper bound $\theta_{\max}$; otherwise it is unfiltered. The set of filtered points $\overline{\mathscr{F}}$ is denoted as:

$$\overline{\mathscr{F}} = \bigcup_{x\in\mathscr{F}} \{y : x \prec y\} \cup \{y : \theta(y) \geq \theta_{\max}\}.$$

Based on a set (population) of points, at a current iteration, the filter-based AFS algorithm generates a set of trial points that would be selected if they improve over the corresponding current points. Then, the best point of the population is selected and would be rejected by the filter if it is a filtered point; otherwise, the best point is not dominated by any other point in the filter and is accepted and added to $\mathscr{F}$. Cumulatively, all trial points that are non-dominated are added to the filter. We note that when adding a point to the filter, all the dominated points are removed from the filter. Thus, at each moment, the filter in the AFS algorithm contains a set of non-dominated points, including the best point, that have been selected from the population along the iterative process.

To avoid the acceptance of a point, or the corresponding pair (θ, f), that is arbitrary close to the boundary of $\overline{\mathscr{F}}$, acceptability of y to the filter requires that

$$\theta(y) \leq \theta(x) - \alpha_1 \text{ or } f(y) \leq f(x) - \alpha_2, \text{ for all } x \in \mathscr{F}$$

for sufficiently small fixed positive parameters α_1, α_2.

As required by the theory, we will prove that the sequence of best points generated by the filter-based AFS algorithm and accepted by the filter will converge to a $(\rho^{(k)}, \varepsilon^{(k)})$-global solution of subproblem (4). Since the solution of each subproblem (4) is obtained by a stochastic method that generates a population of points at each iteration, the position of a point in the population is considered a stochastic vector. Thus, the convergence analysis of the properties of the algorithm relies on the stochastic convergence from the probability theory.

We will now describe the filter-based AFS algorithm and state the conditions for the selection of a non-dominated best point from a population of points. When two points of the population are compared to see which one improves over the other, the below definition is used.

Definition 4 (Point y improves over point x) Let x and y be two points in Ω. The point y improves over point x if one of the following conditions holds:

$$\theta(y) \leq \theta(x) - \alpha_1 \text{ or } \left(f(y) \leq f(x) - \alpha_2 \text{ if } |\theta(y) - \theta(x)| \leq \alpha_{tol}\right), \tag{6}$$

for a small positive parameter α_{tol}.

Restricting the conditions to cases where improvement in θ or in f of more than α_1 and α_2 respectively occurs should not be considered a limitation. These error bounds may be kept as small as one requires[1].

In the context of solving subproblem (4), let f_{best} denote the function value and θ_{best} the constraint violation value of the best point, denoted by x_{best}, of a population of m points, X, at iteration t, where t is the iteration counter of the inner cycle. The best point of a population is selected as follows.

Definition 5 (Best point of a population) The best point of a population $X \in \Omega^m$, is the point x_{best} that satisfies one of the two following conditions:

1. if there are some feasible points in X, x_{best} is the point with least objective function value among the feasible points that is not dominated by the filter:

$$\theta_{\text{best}} = 0 \text{ and } f_{\text{best}} < f(x) \text{ for all } x \in X \text{ such that } \theta(x) = 0 \text{ and } x_{\text{best}} \notin \overline{\mathscr{F}}; \tag{7}$$

2. otherwise, x_{best} is the non-dominated point with least constraint violation among the infeasible points:

$$0 < \theta_{\text{best}} < \theta(x) \text{ and } x_{\text{best}} \notin \overline{\mathscr{F}}. \tag{8}$$

[1] We note that Theorem 3 in Section 3.6 is invalid for $\alpha_1 = \alpha_2 = 0$ because the probability of a stochastic algorithm exactly hitting the global is zero.

3.1 Generating trial points

At each iteration t of the inner cycle, a population of m solutions, herein denoted by $X = [x_1 \ldots x_m]$ is used to generate a set of trial points $Y = [y_1 \ldots y_m]$. When $t = 0$, a set of points is randomly generated in the entire search space Ω using the equation: $(x_j)_i = (l)_i + \xi\,((u)_i - (l)_i)$ for each component $i = 1,\ldots,n$ of the point x_j, where ξ is a uniformly distributed random variable in $[0,1]$. Each fish/point x_j movement is defined according to the number of points inside its 'visual scope'. The 'visual scope' is defined as the closed neighborhood centered at x_j with a positive radius δ. In this study, the radius varies with the point progress and we set as a fraction of the maximum distance between x_j and the other points x_l, $l \neq j$, $\delta_j = \max_l \left\|x_j - x_l\right\|$. Three possible situations may occur:

i) the 'visual scope' is empty;
ii) the 'visual scope' is crowded;
iii) the 'visual scope' is not crowded.

When the 'visual scope' is empty, a Random Behavior is performed, in which the trial y_j is randomly generated inside the 'visual scope' of x_j. The pseudo-code for the procedure Random Behavior is presented in Algorithm 2.

Algorithm 2 (Random Behavior algorithm)

```
Require: $x_j$, $\delta_j$, $0 < \gamma_\delta < 1$;
set $\Delta = \gamma_\delta \delta_j$;
for all $(x_j)_i$, $i = 1,\ldots,n$ do
    set $\xi_1, \xi_2$ uniformly distributed variables in $(0,1]$;
    if $\xi_1 > 0.5$ then
        set $(y_j)_i = (x_j)_i + \xi_2 \Delta$;
    else
        set $(y_j)_i = (x_j)_i - \xi_2 \Delta$;
    end if
end for
return $y_j$;
```

When the 'visual scope' is crowded, with more than 80% of the population inside the 'visual scope' of x_j, a point is randomly selected from the visual, x_{rand}. Then, if it improves over x_j according to the Definition 4, the Searching Behavior is implemented, i.e., y_j is randomly generated along the direction from x_j to x_{rand}. Otherwise, the Random Behavior is performed. The procedure Searching Behavior can be represented by the Algorithm 3.

Algorithm 3 (Searching Behavior algorithm)

```
Require: $x_{\text{rand}}$, $x_j$;
for all $(x_j)_i$, $i = 1,\ldots,n$ do
    set $\xi$ uniformly distributed variable in $(0,1]$;
    set $(y_j)_i = (x_j)_i + \xi((x_{\text{rand}})_i - (x_j)_i)$;
end for
return $y_j$;
```

When the 'visual scope' is not crowded, and the best point inside the 'visual scope', $x_{\min}$, improves over x_j (Definition 4), the Chasing Behavior is performed. This means that

y_j is randomly generated along the direction from x_j to $x_{\min}$. However, if $x_{\min}$ does not improve over x_j, the Swarming Behavior could be tried instead. The central point of the 'visual scope', $\bar{x}$, is computed and if it improves over x_j, y_j is computed randomly along the direction from x_j to $\bar{x}$; otherwise, a point x_{rand} is randomly selected from the 'visual scope'. Then, if this random point improves over x_j the Searching Behavior is implemented, otherwise a Random Behavior is performed. Both Chasing Behavior and Swarming Behavior can be also represented by Algorithm 3, replacing x_{rand} by $x_{\min}$, in the Chasing Behavior case, and by $\bar{x}$, in the other case. The reader is referred to [50,51] for details concerning the AFS algorithm.

We note that each point x_j will produce a trial point defined component by component by

$$(y_j)_i = (x_j)_i + \xi\,(v_j)_i\,, i = 1,\ldots,n \tag{9}$$

where ξ is a uniformly distributed random variable in $(0,1]$ and v_j is a search direction. When a trial point y_j does not satisfy the bound constraints, a projection onto the set Ω is implemented: $(y_j)_i = \max\left\{(l)_i,\, \min\{(y_j)_i,(u)_i\}\right\}$ for each $i = 1,\ldots,n$.

3.2 'Selection Procedure'

To choose which point between the current x_j and the trial y_j will be the point of the population for the next iteration, a 'Selection Procedure' is carried out. The 'Selection Procedure' in this filter-based AFS algorithm is a deterministic process. The current and the trial points are compared with each other and if the trial improves over the current point, decreasing the measure of infeasibility by more than α_1, or the objective function f by more than α_2 without changing θ more than α_{tol} (see Definition 4), the trial point is passed to the next iteration as a current point; otherwise, the current point is preserved to the next iteration.

Since the best point of the current iteration, as well as all the points previously added to the filter that remain in the population, play an essential role in the convergence proof, they are treated separately. The goal is to prevent moving them to positions in the set $\overline{\mathscr{F}}$. All non-dominated points that remain in the population will only be moved to unfiltered improved positions. Hence, each of the corresponding trial point y is passed to the next iteration only if it improves over the current point, according to Definition 4, and cumulatively if it is not dominated by the current filter. Otherwise, the current point is preserved to the next iteration. Algorithm 4 presents the pseudo-code of the 'Selection Procedure'.

Algorithm 4 ('Selection Procedure' algorithm)

```
Require: $x_j, y_j$;
if $x_j$ is the best point $\vee$ $x_j \notin \overline{\mathscr{F}}$ then
    if $y_j$ improves over $x_j$ $\wedge$ $y_j \notin \overline{\mathscr{F}}$ then
        set $x_j = y_j$;
    end if
else
    if $y_j$ improves over $x_j$ then
        set $x_j = y_j$;
    end if
end if
return $x_j$;
```

3.3 Best point

After a set of points has been selected to define the population for the next iteration, the best point is required. At each iteration, x_{best} is the best point among all the points of the population that is non-dominated by the current filter, in the sense of Definition 5. This best point is accepted and added to the current filter. The sequence of best points thus found will converge to the $(\rho^{(k)}, \varepsilon^{(k)})$-global minimizer $x^{(k)}$ of subproblem (4).

However, when a non-dominated best point is not found and before declaring the iteration *unsuccessful*, it is required that a restoration phase with a local search about a reference point in the filter be carried out to see if a non-dominated point that improves θ or f relative to the reference point is found. Only after this local search has failed can an iteration be declared *unsuccessful*.

3.4 Restoration phase: local search

When it is not possible to find a non-dominated best point, a restoration phase is invoked. Here, the most nearly feasible point in the filter, $x_{\mathscr{F}}^{\inf}$, i.e., the point $x_{\mathscr{F}}^{\inf} \in \mathscr{F}$ such that $\theta(x_{\mathscr{F}}^{\inf}) \leq \theta(x)$ holds for each $x \in \mathscr{F}$, is recuperated and a local coordinate search is carried out about it. We note that $x_{\mathscr{F}}^{\inf}$ may be feasible. θ and f are calculated at a set of points along the coordinate axes with a random step size $\xi\sigma$ where ξ is a uniformly distributed variable in $(0,1]$ and $\sigma > 0$, expressed in the form

$$y_r = x_{\mathscr{F}}^{\inf} + \xi\sigma d_r,\ r = 1,\ldots,2n \tag{10}$$

where $d_r, r = 1,\ldots,2n$ are the columns of the matrix $D = [e_1 \cdots e_n - e_1 \cdots - e_n]$, being e_i the ith unit coordinate vector. We note that this search sets $\theta(y_r) = \infty$ if $y_r \notin \Omega$, to maintain the search inside the bounds. To adapt the local search to each problem, we set similarly to [30]

$$\sigma = \max\left\{10^{-5}, \min\left\{\sigma_{\min}, 0.05\frac{\sum_{i=1}^{n}(u)_i - (l)_i}{n}\right\}\right\},$$

where $\sigma_{\min}$ is reduced 10% at every iteration of the outer cycle to suit the progress of the algorithm.

Let y' be the best of these $2n$ points y_r, in the sense of Definition 5. If it is possible to find a non-dominated best point y' among these $2n$ points, then sufficient progress in θ or in f is verified, relative to the point $x_{\mathscr{F}}^{\inf}$:

$$\theta(y') \leq \theta(x_{\mathscr{F}}^{\inf}) - \alpha_1 \text{ or } f(y') \leq f(x_{\mathscr{F}}^{\inf}) - \alpha_2, \tag{11}$$

the point y' is added to the filter and replaces the best point of the population at the current iteration. We note that when it is not possible to find a non-dominated best point y', the iteration is declared *unsuccessful*.

3.5 The filter-based AFS algorithm

In this filter-based AFS algorithm with the incorporated filter method, each iteration is declared either successful or *unsuccessful*. A successful iteration is the one where a non-dominated best point is found, or the one where a non-dominated best point is not found, but the restoration phase produces a point y' that is accepted to the filter, thus satisfying conditions (11). When an unfiltered best point is not found and, in the restoration phase, the local coordinate search about $x_{\mathscr{F}}^{\inf}$ could not find a point y' that is acceptable to the filter, the iteration is declared *unsuccessful*. Here the best point of the previous iteration is preserved to the next one.

At the beginning of the algorithm, the set of filtered points is initialized to

$$\overline{\mathscr{F}}_0 = \{x : x_{\text{best}} \prec x\} \cup \{x : \theta(x) \geq \theta_{\max}\}$$

meaning that the approximation x_{best} of the previous iteration in the outer cycle is added to the filter $\mathscr{F}$, which implies that this point is a filtered point at the current iteration, and the filter will never accept points that have a constraint violation larger than $\theta_{\max}$. In the initialization of $\overline{\mathscr{F}}_0$, $\theta_{\max}$ is a large multiple of the infeasibility measure of the best point targeting the exploration of the whole search space for a global solution: $\theta_{\max} = 10^4 \max\{1, \theta_{\text{best}}\}$. Furthermore, whenever a point y is accepted to the filter $\mathscr{F}$, the set of filtered points is updated as follows:

$$\overline{\mathscr{F}}_{t+1} = \overline{\mathscr{F}}_t \cup \{x : y \prec x\},$$

where we note that all accepted points, even the feasible ones, are included in the filter. When a point y is included in the filter, all points that are dominated by y are removed from the filter.

The pseudo-code for the filter-based AFS algorithm is presented below in Algorithm 5. The output point, x_{best}, returned by this algorithm is the required $(\rho^{(k)}, \varepsilon^{(k)})$-global minimizer $x^{(k)}$ of the Algorithm 1. We note that if a point that satisfies the stopping conditions is not found within $t_{\max}$ iterations, Algorithm 5 is stopped and the best point in the filter will be the initial approximation for the next subproblem.

3.6 Convergence study of the filter-based AFS algorithm

In the previous section, convergence to an ε-global minimizer of problem (1) has been guaranteed, provided that the subproblems are $(\rho^{(k)}, \varepsilon^{(k)})$-globally solved, where $\rho^{(k)} \to 0$ and $\varepsilon^{(k)} \to \varepsilon$. Thus, the issue here is to guarantee that a $(\rho^{(k)}, \varepsilon^{(k)})$-global minimizer $x^{(k)}$ of subproblem (4) is found by the Algorithm 5 so that conditions (5) are satisfied for all $x \in \Lambda \subseteq \Omega$. Since the set Ω is compact and both $f(x)$ and $\theta(x)$ are continuous, a $(\rho^{(k)}, \varepsilon^{(k)})$-global minimizer of subproblem (4) does exist. Hence, conditions for which the filter-based AFS algorithm converges to a $(\rho^{(k)}, \varepsilon^{(k)})$-global optimizer are herein discussed.

Assuming that the objective function $f(x)$ of subproblem (4) is bounded below for all $x \in \Omega$, the main convergence results of the filter-based AFS algorithm are the following. To simplify the notation, the dependence on k will be dropped from $\rho^{(k)}$ and $\varepsilon^{(k)}$ during the remaining part of this section. The below provided convergence analysis is an extension of that established in [27] for a particular class of evolution strategies when solving unconstrained problems, and it is adapted to a filter set method in which the concept of unfiltered points is crucial. The analysis makes use of the probability theory from the measure theoretic point of view. The reader is referred to the recent textbook [20] for details. We remark that in the AFS algorithm, all points of the population are potentially modified by one of the fish behavior, in contrast to some evolutionary algorithms which often select only a subset of points to apply the evolutionary operators. On the other hand, the application of fish behavior to all points increases diversity. The ‘Selection Procedure’, the concept of best point of a population and the restoration phase, all aiming to identify unfiltered points, are crucial to promote convergence.

Algorithm 5 (filter-based AFS algorithm)

```
Require: $\rho^{(k)}$, $\varepsilon^{(k)}$, $x_1$ ($x^{(k-1)}$ from Algorithm 1), $LB$;
set $t = 0$;
initialize $\mathscr{F}$;
randomly generate $m-1$ points in $\Omega$: $x_j, j = 2,\dots,m$;
select $x_{\text{best}}$ using Definition 5;
while $\theta_{\text{best}} > \rho^{(k)} \;\vee\; f_{\text{best}} > LB + \varepsilon^{(k)}$ do
    for all $x_j$, $j = 1,\dots,m$ do
        compute 'visual scope';
        if 'visual scope' is empty then
            Random Behavior;
        else if 'visual scope' is crowded then
            if $x_{\text{rand}}$ improves over $x_j$ then
                Searching Behavior;
            else
                Random Behavior;
            end if
        else if $x_{\min}$ improves over $x_j$ then
            Chasing Behavior;
        else if $\bar{x}$ improves over $x_j$ then
            Swarming Behavior;
        else if $x_{\text{rand}}$ improves over $x_j$ then
            Searching Behavior;
        else
            Random Behavior;
        end if
        'Selection Procedure';
        update $\mathscr{F}$ when appropriate;
    end for
    if a new best point is found based on Definition 5 then
        update $x_{\text{best}}$;
    else if a non-dominated point $y'$ satisfying (11) is found then
        $x_{\text{best}} \leftarrow y'$;
    else
        preserve $x_{\text{best}}$;
    end if
    set $t = t + 1$;
end while
return $x_{\text{best}}$;
```

We consider the problem of finding a (ρ,ε)-global minimizer of the bi-objective optimization problem (4). We assume that the following conditions hold:

Assumption A 6 *Assume that:*

i) the search space Ω is Lebesgue measurable;

ii) the functions $f:\Omega\to\mathbb{R}$ and $\theta:\Omega\to\mathbb{R}_+$ are measurable functions;

iii) a global minimizer of $f(x)$ that satisfies $h(x)=0$ and $g(x)\le 0$ exists in Ω;

iv) if $m(S)$ denotes the Lebesgue measure of a set S, then for any pair of positive parameters (ρ,ε),

$$m\left(\left\{x:\theta(x)\le\rho,\, f(x)\le \min_{z\in\Lambda\subseteq\Omega} f(z)+\varepsilon\right\}\right) > 0. \tag{12}$$

Conditions i) – iv) are satisfied as long as Ω is a bounded region and f and θ are continuous. It seems natural that a stochastic search algorithm works only if the probability of the points generated by the algorithm intersecting the set $\{x:\theta(x)\le\rho,\, f(x)\le \min_{z\in\Lambda\subseteq\Omega} f(z)+\varepsilon\}$

is greater than zero, for any positive parameters ρ and ε. Thus, the concept of size or measure with respect to a set is required. The Lebesgue measure is a natural framework for the study of the convergence properties of the filter-based AFS algorithm that follows [20].

The convergence of the filter-based AFS algorithm is expressed in the following theorem.

Theorem 3 *Let θ and f be two objective functions defined over a search space Ω. If the pair (θ, f) and Ω satisfy Assumption A 6, then for $\rho, \varepsilon > 0$ the filter-based AFS algorithm, as defined by Algorithm 5, will converge to a (ρ, ε)-global minimizer of problem* (4) *with probability one, in the sense that conditions* (5) *are satisfied for a best point of the population that is acceptable to the filter.*

Proof We first assume that the population has just one point ($m = 1$), $x \equiv x_{\text{best}} \in \Omega$. The definition of best point of a population and the 'Selection Procedure' play here important roles, since the trial point y that is generated by equation (9) is passed to the next iteration as a current point if it is not dominated by the filter[2] and a decrease in constraint violation by more than α_1, or in objective function value by more than α_2 without changing θ more than α_{tol}, is verified. Furthermore, y will the best point because it is not dominated by the current filter and Definition 5 then applies.

However, when the trial y is not accepted, the current x could not be x_{best} since x has been added to the filter in a previous iteration. The filter methodology in Algorithm 5 invokes the restoration phase. The local search procedure finds the best point y' that is acceptable and added to the current filter, satisfying (11), thus replacing the current x_{best}. Consequently, each successful iteration improves either the infeasibility measure (decreases θ) by more than α_1 or the objective function value (decreases f) by more than α_2, and cumulatively a non-dominated point is generated following Definition 5. Hence, an approximate global minimizer can always be reached from any initial point after a finite number of successful iterations.

We note that if the local search does not find a point y' that is acceptable to the current filter, the current x_{best} is preserved to the next iteration and an *unsuccessful* iteration occurs.

Let the set S be defined as

$$S = \{y \in \Omega \backslash \overline{\mathscr{F}} : \theta(y) \leq \rho, f(y) \leq f(x) + \varepsilon, \text{ for all } x \in \Lambda \subseteq \Omega\} \subset \Omega \backslash \overline{\mathscr{F}}$$

and p denote the probability that an unfiltered point y satisfies the conditions (5), i.e., $p \equiv Prob[y \in S]$. Then, by item iv) in Assumption A 6, $\mathtt{m}(S) > 0$. We note that the uniform distribution of ξ, when generating a trial point y, as described in (9), or y', as described in (10), together with $\mathtt{m}(S) > 0$, imply $p > 0$[3].

We show that if x is not an approximate global minimizer, then with probability one a successful iteration occurs within a finite number of iterations. The proof is by contradiction, assuming that an *unsuccessful* iteration occurs infinitely many times. The probability that an

[2] Recall that the current point has been surely added to the filter in a previous iteration.

[3] Any other distribution with density function that is nowhere zero over the set Ω satisfies this requirement.

unsuccessful iteration occurs at iteration t is given by[4]

$$\begin{aligned} q_t &= Prob\left[\text{ `}y\text{ does not improve over }x_{\text{best}}\text{' or } y \in \overline{\mathscr{F}}\,\right] Prob\left[y' \in \overline{\mathscr{F}}\,\right] \\ &= Prob\left[y \in \overline{\mathscr{F}}\,\right] Prob\left[y' \in \overline{\mathscr{F}}\,\right] \\ &= \left(1 - Prob\left[y \in \Omega \backslash \overline{\mathscr{F}}\,\right]\right)\left(1 - Prob\left[y' \in \Omega \backslash \overline{\mathscr{F}}\,\right]\right) \\ &< (1 - Prob[y \in S])^2 = (1-p)^2 \\ &< 1. \end{aligned}$$

When we assume that x is preserved, p remains unchanged during the iterative process. Since each iteration is independent, the probability of an *unsuccessful* iteration occurring during r consecutive iterations is given by

$$q_1 q_2 \cdots q_r = \prod_{t=1}^{r} q_t \leq (1-p)^{2r}$$

and the probability of occurring infinitely many times is

$$\lim_{r \to \infty} \prod_{t=1}^{r} q_t = 0.$$

Thus, we may conclude that if x is not an approximate global minimizer, then within a finite number of iterations a non-dominated point is guaranteed to be generated. Thus, after a finite number of non-dominated points, an approximate global solution is reached.

Now, we address the case of a population with $m > 1$ points. We remark that when a method based on a population of size m, X, is used, the search space is Ω^m. For the convergence analysis of Algorithm 5, we are interested in the sequence of points, $x_{\text{best}} \in X$, that are non-dominated following Definition 5.

In this context, we re-define the set S as follows:

$$S = \{Y \in \Omega^m \backslash \widetilde{\overline{\mathscr{F}}} : \theta_{\text{best}} \leq \rho,\, f_{\text{best}} \leq f(x) + \varepsilon,\ \text{for all}\, x \in \Lambda \subseteq \Omega\} \subset \Omega^m \backslash \widetilde{\overline{\mathscr{F}}}$$

where

$$\widetilde{\overline{\mathscr{F}}} = \bigcup_{x \in \mathscr{F}} \{X : x \prec x_{\text{best}}\} \cup \{X : \theta_{\text{best}} \geq \theta_{\max}\}$$

represents the set of populations in which x_{best} is a point that is dominated by the filter and $\widetilde{p} \equiv Prob[Y \in S]$ is the probability of a non-dominated best point of a population Y with m points be an approximate global minimizer.

We note that when the trial point generated from the best point of the current iteration is passed to the next iteration, as described in Section 3.2, no guarantee exists that this will be the best point of the next iteration. Some other point of the new population could be better

[4] When $m = 1$, an *unsuccessful* iteration occurs when the events A and B both occur, where

A: the trial point y is not accepted, either because it does not improve over the current $x \equiv x_{\text{best}}$ or $y \in \overline{\mathscr{F}}$;
B: in the restoration phase, a non-dominated best point is not found among the $2n$ points generated by (10).

Hence $Prob\,[\textit{unsuccessful}\text{ iteration}] = Prob\,[A \cap B] = Prob\,[A]\, Prob\,[B|A] = Prob\,[A]\, Prob\,[B]$ since A and B are independent. We further note that the event 'y does not improve over x_{best}' implies '$y \in \overline{\mathscr{F}}$'. This means that if the events A_1 and A_2 are

A_1: y does not improve over x_{best};
A_2: $y \in \overline{\mathscr{F}}$;

then $Prob\,[A_1 \cup A_2] = Prob\,[A_2]$ since $A_1 \subset A_2$.

in θ or in f than the trial of the current best point, according to the Definition 5. Either way, a new non-dominated best point is found and the iteration is successful.

However, if the trial point, obtained from the current best point, is not accepted, meaning that it does not improve over the best point or it is dominated by a point in the current filter, then some other point in the next population should satisfy Definition 5 so that the iteration could be declared successful. If no other point in the next population satisfies Definition 5, the restoration phase is invoked and the procedure is similar to the one described in the case of a population with $m = 1$. We may conclude that each successful iteration generates a non-dominated best point that improves either θ or f, as stated in Definition 5. Thus, after a finite number of successful iterations, an approximate global minimizer can always be reached.

We will now show that if x_{best} of the current population is not an approximate global minimizer, then with probability one a successful iteration occurs within a finite number of iterations. Again, by contradiction assume that an *unsuccessful* iteration occurs infinitely many times. Let q_t be the probability that an *unsuccessful* iteration occurs at iteration t. Then[5]

$$\begin{aligned} q_t &= Prob\left[y_{\text{best}} \in \overline{\mathscr{F}}\right] Prob\left[y' \in \overline{\mathscr{F}}\right] \\ &= \left(1 - Prob\left[y_{\text{best}} \in \Omega \backslash \overline{\mathscr{F}}\right]\right)\left(1 - Prob\left[y' \in \Omega \backslash \overline{\mathscr{F}}\right]\right) \\ &= \left(1 - Prob\left[Y \in \Omega^m \backslash \widetilde{\overline{\mathscr{F}}}\right]\right)^2 \\ &< (1 - Prob[Y \in S])^2 = (1 - \widetilde{p})^2 \\ &< 1, \end{aligned}$$

where $Prob[Y \in S] > 0$. From here now, the same reasoning applies to this case to conclude that if the best point of the population, at the current iteration, is not an approximate global minimizer, then within a finite number of iterations a new population with a non-dominated best point is guaranteed to be generated. Therefore, after a finite number of iterations where the best point of the population is not dominated by the filter, an approximate global minimizer is reached. □

Remark 1 These convergence results would apply to any stochastic population-based algorithm, provided that each point x, of the current population, generates a trial point $y \neq x$, and the trial replaces the current point for the next population if at least it improves over the current according to Definition 4. Furthermore, the best point of each population should be identified as described in Definition 5. All the remaining properties are concerned with the developed filter methodology.

4 Numerical experiments

For a preliminary validation of the proposed population-based filter algorithm two sets of benchmark constrained global optimization problems are used. First, we compare the results of this study with those in [19], that were obtained by a non-differentiable exact penalty function framework using the deterministic DIRECT algorithm for globally solving the bound constrained subproblems. For this comparison, a set of 20 small problems (fully described in the report [9], also available in `http://www.ime.usp.br/~egbirgin/`) is

[5] When $m > 1$, an *unsuccessful* iteration occurs when the events A and B both occur, where

A: a non-dominated best point y_{best} is not found in the next population;
B: the local search, in the restoration phase, does not find a point y' that is acceptable to the filter;

and again, these two events are independent.

used. These problems are identified by Problem 1, 2(a), 2(b), 2(c), 2(d), 3(a), 3(b), 4, 5, 6, ..., 15, 16 in [9,10,19]. Second, we compare the performance of the proposed method with another filter based technique [30], therein denoted by 'FSA'. In this last paper, the heuristic simulated annealing algorithm is coupled with a multistart diversification strategy, and an intensification scheme to accelerate convergence is also used. We further extend our numerical testing to compare with a population-based stochastic electromagnetism-like (EM) method, therein called 'CEM'. The extension of the EM concepts to constrained global optimization problems relies on computing the charge of each point in the population by combining two charge components [3]. The first component is due to the objective function value and the second is due to the constraint violation. The idea of using the function value and constraint violation separately is present but the method itself is not a filter method. For these last comparisons, a set of 13 benchmark test problems with dimensions ranging from 2 to 20, identified as g01, g02, g03, g04, ..., g12, g13 (fully described in [30]) is used. Although there are maximization problems in this set, they were formulated and solved as minimization problems. The characteristics of the problems are summarized in Table 1 that lists the number of variables, 'n'; the type of objective function, 'type of f'; the number of linear inequalities, 'l. ineq.'; the number of nonlinear inequalities, 'nl. ineq.'; the number of linear equalities, 'l. eq.'; and nonlinear equalities, 'nl. eq.'. Columns under '**P**' identify the problem. The C++ programming language is used in this real-coded algorithm. The computational tests were performed on a PC with a 2.8 GHz Core Duo Processor P9700 and 6 Gb of memory. Since the algorithm relies on some random parameters and variables, we solve each problem 30 times. The parameters have been set as follows after an empirical study: $\gamma_p = 0.1$, $\gamma_\varepsilon = 0.1$, $\rho^{(1)} = 1$, $\varepsilon^{(1)} = 10$, $\varepsilon = 10^{-4}$, $\rho_{tol} = 10^{-8}$, $\gamma_\delta = 0.8$, $\alpha_1 = \alpha_2 = 10^{-8}$ and $\alpha_{tol} = 0.001$. The initial value for $\sigma_{\min}$ is set to 10.

Table 1 Test problems

P	n	type of f	l. ineq.	nl. ineq.	l. eq.	nl. eq.	**P**	n	type of f	l. ineq.	nl. ineq.	l. eq.	nl. eq.
1	5	polynomial	0	0	0	3	14	4	nonlinear	2	0	1	0
2(a)	9	linear	0	2	3	1	15	3	constant	0	0	1	2
2(b)	9	linear	0	2	3	1	16	5	linear	0	0	1	2
2(c)	9	linear	0	2	3	1	g01	13	quadratic	9	0	0	0
2(d)	10	linear	0	2	4	1	g02	20	nonlinear	0	2	0	0
3(a)	6	linear	0	1	0	4	g03	10	polynomial	0	0	0	1
3(b)	2	nonlinear	0	1	0	0	g04	5	quadratic	0	6	0	0
4	2	linear	0	1	0	0	g05	4	cubic	2	0	0	3
5	3	linear	0	0	0	3	g06	2	cubic	0	2	0	0
6	2	linear	0	1	0	0	g07	10	quadratic	3	5	0	0
7	2	linear	2	2	0	0	g08	2	nonlinear	0	2	0	0
8	2	polynomial	1	1	0	0	g09	7	polynomial	0	4	0	0
9	6	nonlinear	3	0	3	0	g10	8	linear	3	3	0	0
10	2	linear	0	2	0	0	g11	2	quadratic	0	0	0	1
11	2	quadratic	0	1	0	0	g12	3	quadratic	0	1	0	0
12	2	quadratic	0	0	0	1	g13	5	nonlinear	0	0	0	3
13	3	nonlinear	0	0	1	1							

Table 2 aims to compare the results obtained by our P-BF AFS-based algorithm and those reported in [19]. For this comparison, the size of the population is defined as $m = \min\{50, 5n\}$, $t_{\max} = 20$ and Algorithm 1 is allowed to run for a maximum of 10 iterations. The table identifies the problem in the first column under '**P**', and reports the best solution obtained during the 30 runs[6], 'f_{best}'; the median of the 30 solutions[7], 'f_{median}'; the solution found by the algorithm in [19], 'f^*'; and the best known solution available in the literature, '*LB*' [10]. Furthermore, the measure of the constraint violation, 'C.V.'[8], the number of function evaluations, 'n.f.e.', and the CPU time in seconds, 'time', (reported in the table) are from the best solutions obtained by the algorithms, i.e., correspond to f_{best} in P-BF AFS-based algorithm, f^* in [19] and *LB* in [10]. We note that all the results used for comparison are taken from their original references [3,10,19,30].

For the Problems 2(a), 2(b) and 2(c), some algebraic manipulation aiming to reduce the number of variables and equality constraints is used to reformulate them [19]. The new formulation has five variables and 10 inequality constraints. A similar strategy is used to obtain a new formulation of Problem 2(d) with five variables and 12 inequality constraints [19]. We further extended this idea to other difficult to solve problems in the set. Considering the new formulations, Problem 5 has now two variables, two equality constraints and two inequality constraints; Problem 9 has three variables and nine inequality constraints; Problem 12 has one variable and two inequality constraints; Problem 14 has three variables and four inequality constraints and Problem 16 has two variables and six inequality constraints. As a consequence, the computational effort to solve these problems has drastically been reduced.

From the results we may conclude that the proposed P-BF AFS-based algorithm is able to reach the target solution with high accuracy, except in Problem 1 and Problems 2(a), 2(b) and 2(c). We conclude that the P-BF AFS-based algorithm reached the '*LB*' solution in 16 problems out of 20, while the algorithm in [19] only reached the best known solution in 10 problems. The consistency of our results is very good with f_{best} very near f_{median}, with the exceptions of Problems 1, 2(a), 2(b), 2(c) and 13. The algorithm is able to reach feasible solutions. Only in two of the 20 problems (Problem 1 and Problem 3(a)) the C.V. of our reported solution is larger than that obtained in [19]. Although different computational platforms have been used to solve the selected problems, the reported values of 'time' required by P-BF AFS-based algorithm are very competitive when compared with those in [19]. The computational effort measured by 'n.f.e.' are encouraging for a population-based stochastic method, when compared with the values listed in [19].

We now compare the results obtained by P-BF AFS-based algorithm with those in [30]. The results summarized in Table 3 show, for each problem, the best, 'f_{best}', and the average, 'f_{avg}', solutions obtained in 30 independent runs, as well as the standard deviation, 'St.D.', and the 'C.V.' of the best solution. Moreover, the average number of function evaluations (over the 30 runs), 'n.f.e.$_{\text{avg}}$', and the average number of constraint evaluations, 'n.c.e.$_{\text{avg}}$', are also reported for comparison. We remark that n.c.e.$_{\text{avg}}$ = n.f.e.$_{\text{avg}}$ in P-BF AFS-based algorithm. A population of 50 points is used. FSA also uses 50 points in the set from which points may be chosen to generate other solutions, within the multistart diversification strategy. Each algorithm is terminated by using its own specific stopping conditions. Values of $\varepsilon = 10^{-4}$ and $\rho_{tol} = 10^{-8}$ are maintained in the stopping conditions of Algorithm 1 which is allowed to run for a maximum of 10 iterations. To allow good solutions to be found, we set during these experiments $t_{\max} = 200$. All the other parameters are set as previously described. The best known objective function values available in the literature for the set of test problems g01 – g13, '*LB*', are listed in Table 4. From the comparison with FSA, we

[6] For P-BF AFS-based algorithm, the reported solution is the best solution obtained during the 30 runs. During selection, priority is given to the constraint violation. Among all feasible solutions, the best is the one with smallest objective function value. Among infeasible solutions, the best is the one with smallest constraint violation.

[7] This is a measure of central tendency of the distribution of the 30 obtained solutions.

[8] For comparative purposes, the measure of C.V. used in the table is $\max\{\|h(x)\|_\infty, \|g(x)_+\|_\infty\}$ [19].

Table 2 Comparison between P-BF AFS-based algorithm and the non-differentiable exact penalty in [19]

	P-BF AFS-based algorithm					results in [19]				in [10]	
P	f_{best}	f_{median}	C.V.	n.f.e.	time	f^*	C.V.	n.f.e.	time	*LB*	time
1	0.0956	1.4665	7.84e-07	6 945	0.031	0.0625	2.35e-07	39 575	0.328	0.0293	18.86
2(a)	-358.650	-308.664	0.00e00	7 068	0.046	-134.113	8.43e-04	115 107	2.078	-400.000	0.13
2(b)	-378.317	-274.472	0.00e00	6 963	0.046	-768.457	5.30e-04	120 057	3.828	-600.000	0.76
2(c)	-697.452	-657.349	0.00e00	7 189	0.046	-82.977	8.43e-04	102 015	0.953	-750.000	0.16
2(d)	-399.118	-394.563	0.00e00	6 526	0.031	-385.170	0.00e00	229 773	2.328	-400.000	0.23
3(a)	-0.3888	-0.3842	5.22e-04	7 495	0.046	-0.3861	1.02e-06	48 647	1.234	-0.3888	12.07
3(b)	-0.3888	-0.3888	0.00e00	1 041	0.000	-0.3888	0.00e00	3 449	0.031	-0.3888	2.90
4	-6.6667	-6.6665	0.00e00	493	0.000	-6.6666	0.00e00	3 547	0.031	-6.6666	0.00
5	201.159	201.159	8.11e-07	2 999	0.015	201.159	1.66e-04	14 087	0.078	201.16	0.04
6	376.293	376.304	0.00e00	1 335	0.000	0.4701	2.05e-05	1 523	0.000	376.29	0.01
7	-2.8284	-2.8283	0.00e00	920	0.000	-2.8058	0.00e00	13 187	0.125	-2.8284	0.02
8	-118.704	-118.698	0.00e00	1 521	0.000	-118.704	0.00e00	7 621	0.046	-118.70	0.15
9	-13.4018	-13.4007	0.00e00	1 839	0.015	-13.4026	1.35e-04	68 177	2.171	-13.402	0.00
10	0.7418	0.7418	0.00e00	2 126	0.015	0.7420	0.00e00	6 739	0.078	0.74178	0.01
11	-0.5000	-0.5000	0.00e00	782	0.000	-0.5000	0.00e00	3 579	0.031	-0.5000	0.01
12	-16.7389	-16.7389	0.00e00	35	0.000	-16.7389	5.36e-06	3 499	0.015	-16.739	0.01
13	189.345	253.937	0.00e00	4 031	0.015	195.955	9.21e-04	8 085	0.078	189.35	0.47
14	-4.5142	-4.5139	0.00e00	2 028	0.015	-4.3460	9.22e-05	19 685	0.250	-4.5142	0.00
15	0.0000	0.0000	9.11e-07	3 593	0.015	0.0000	4.94e-05	1 645	0.000	0.0000	0.06
16	0.7049	0.7049	0.00e00	447	0.015	0.7181	2.00e-04	22 593	0.312	0.70492	0.15

may conclude that the average solutions found by the P-BF AFS-based algorithm are better than those of FSA in 6 out of 13 problems, g01, g02, g03, g04, g10 and g13 (see 'f_{avg}' and 'St.D.'). However, the computational effort measured by 'n.f.e.$_{\text{avg}}$+n.c.e.$_{\text{avg}}$' is lower with our algorithm than with FSA in 8 out of the 13 problems. It is noteworthy that the P-BF AFS-based algorithm reaches the best known solution of problem g01, with the chosen tolerance of $\varepsilon = 10^{-4}$, in all 30 runs, reaches the '*LB*' of problem g08 in 18 out of 30 runs, and the '*LB*' of problem g12 in 6 out of 30 runs. For the other problems of the set, the solutions obtained in all 30 runs have absolute errors larger than ε, since 10 outer iterations were exceeded.

Table 3 Comparison between P-BF AFS-based algorithm and FSA in [30]

	P-BF AFS-based algorithm					results of FSA in [30]				
P	f_{best}	f_{avg}	St.D.	C.V.	n.f.e.$_{\text{avg}}$	f_{best}	f_{avg}	St.D.	n.f.e.$_{\text{avg}}$	n.c.e.$_{\text{avg}}$
g01	-14.99999	-14.99992	2.3e-05	0.00e00	48929	-14.99911	-14.99332	4.8e-03	205748	87701
g02	-0.764816	-0.730774	1.8e-02	0.00e00	104312	-0.754913	-0.371708	9.8e-02	227832	101903
g03	-1.000008	-0.999575	4.7e-04	8.32e-06	51994	-1.000000	-0.999187	1.7e-03	314938	118404
g04	-30665.538	-30665.524	1.0e-02	9.84e-05	102188	-30665.538	-30665.467	1.7e-01	86154	37000
g05	5126.4983	5128.4768	1.5e+00	1.15e-05	112853	5126.4981	5126.4981	0.0e00	47661	17757
g06	-6961.8138	-6961.8127	6.2e-04	0.00e00	106718	-6961.8139	-6961.8139	0.0e00	44538	15817
g07	24.63254	25.43844	3.6e-01	0.00e00	117449	24.31057	24.37953	7.1e-02	404501	171299
g08	-0.095825	-0.095824	3.4e-07	0.00e00	8967	-0.095825	-0.095825	0.0e00	56476	23219
g09	680.64912	680.66745	8.6e-03	0.00e00	106406	680.63008	680.63642	1.5e-02	324569	147035
g10	7077.5240	7198.3822	5.1e+01	0.00e00	125880	7059.8635	7509.3210	5.4e+02	243520	93667
g11	0.749900	0.749901	8.1e-07	1.00e-04	75997	0.749999	0.749999	0.0e00	23722	8485
g12	-1.000000	-0.999998	6.5e-07	0.00e00	11494	-1.000000	-1.000000	0.0e00	59355	25818
g13	0.056265	0.289244	1.3e-01	1.52e-03	95508	0.053950	0.297720	1.9e-01	120268	42268

Finally, we compare the results obtained by P-BF AFS-based algorithm with those of CEM proposed in [3]. To match the conditions for the experiments we set a population of 10 points and the algorithms terminate after 350000 function evaluations. Consequently, the number of iterations in the outer cycle of our algorithm, averaged over 30 runs, is over one hundred mostly, except in problems g01, g02, g07 and g10. The results are reported in Table 4 that shows 'f_{best}', 'f_{avg}' and the objective function value of the worst solution obtained during the 30 runs, 'f_{worst}'. The value of 'C.V.' for the best solution is also shown for our algorithm. As noted in the original reference, the results in [3] that are lower than the best known value '*LB*' occur due to the parameter $\beta = 10^{-3}$ that is used to convert equality constraints $h(x) = 0$ into inequalities $|h(x)| - \beta \leq 0$. The results of the table show that the best reached solution, as well as the average of the 30 solutions, found by the P-BF AFS-based algorithm are better than the corresponding solutions found by CEM in 8 out of the 13 problems. Both algorithms behave similarly with the problems g01, g08 and g12. During these experiments, the P-BF AFS-based algorithm is able to reach the '*LB*' of problems g01, g06, g08, g11 and g12, within a tolerance of $\varepsilon = 10^{-4}$, in all 30 runs. Further, it reaches the '*LB*' of problem g03 in 24 out of 30 runs, and the '*LB*' of problem g04 in 5 out of 30 runs. For the other problems, the difference between the obtained solution and the best known solution is larger than ε, in all runs. We note here that the algorithm stops when the number of function evaluations exceeds 350000. We further extended these experiments and run the algorithm to report the runs (among the 30) where the solution reaches the target '*LB*' with a relative precision of 10^{-4}. This condition was verified in all the 30 runs of problems g01, g04, g06, g08, g09, g11 and g12, in 24 runs of problem g03 and in 8 runs of problem g05.

Overall we may conclude that the results obtained by the P-BF AFS-based algorithm are competitive with the ones reported in [3,30].

Table 4 Comparison between P-BF AFS-based algorithm and CEM in [3]

	P-BF AFS-based algorithm				results of CEM in [3]			
P	f_{best}	f_{avg}	f_{worst}	C.V.	f_{best}	f_{avg}	f_{worst}	*LB*
g01	-15.000000	-15.000000	-15.000000	0.00e00	-15.000	-15.000	-15.000	-15.00000
g02	-0.776103	-0.748956	-0.725272	0.00e00	-0.623711	-0.517221	-0.452235	-0.803619
g03	-1.000000	-0.999943	-0.999674	4.79e-09	-1.00151	-1.00167	-1.00176	-1.000000
g04	-30665.539	-30665.538	-30665.537	0.00e00	-30665.513	-30660.649	-30654.500	-30665.539
g05	5126.4117	5128.3672	5131.5444	6.60e-06	5126.4842	5128.6958	5136.6618	5126.4981
g06	-6961.8139	-6961.8139	-6961.8139	0.00e00	-6961.813	-6961.813	-6961.813	-6961.8139
g07	24.62939	25.06516	25.43367	9.98e-06	25.11276	27.75496	29.93511	24.306209
g08	-0.095825	-0.095825	-0.095825	0.00e00	-0.095825	-0.095825	-0.095825	-0.095825
g09	680.63610	680.65222	680.66051	0.00e00	680.8968	681.3511	681.7680	680.63006
g10	7061.6484	7131.2982	7177.8494	0.00e00	7049.758	7154.671	7292.724	7049.3307
g11	0.749990	0.749990	0.749990	9.99e-06	0.7499	0.7499	0.7499	0.750000
g12	-1.000000	-1.000000	-1.000000	0.00e00	-1.0000	-1.0000	-1.0000	-1.000000
g13	0.055416	0.158980	0.276735	4.17e-07	0.053827	0.056314	0.059852	0.0539498

5 Conclusions

We have demonstrated that it is possible to derive a population-based filter algorithm with guaranteed convergence to an ε-global minimum of a CGO problem. The algorithm solves a sequence of bi-objective subproblems with increasing accuracy, namely with error tolerances of $\rho^{(k)}$ in feasibility and $\varepsilon^{(k)}$ in objective function value, where $\rho^{(k)} \to 0$ and $\varepsilon^{(k)} \to \varepsilon$, as iterations proceed ($k \to \infty$). For each subproblem, the approximate solution of the previous subproblem together with a randomly generated population of points aim to explore the search space for a global minimum instead of being trapped in a local one.

In the proposed population-based filter framework, the subproblems are globally solved by an artificial fish swarm stochastic algorithm that makes use of the filter set method to provide selection criteria to accept a sequence of non-dominated best points of the population. The filter-based AFS algorithm convergence to a $(\rho^{(k)}, \varepsilon^{(k)})$-global minimizer of the bi-objective subproblem (4) has been guaranteed using probability theory, where $\rho^{(k)} \to 0$ and $\varepsilon^{(k)} \to \varepsilon$.

The reported preliminary numerical results show its competitive practical performance when compared with a penalty function framework based on the deterministic DIRECT algorithm and with two stochastic heuristic methods, targeting global solutions. Extensive numerical experiments with larger dimensional problems remain to be done and will be reported in a future paper.

Acknowledgements The authors wish to thank the anonymous referees for their fruitful comments and suggestions. This work was financed by FEDER funds through COMPETE (Operational Programme Thematic Factors of Competitiveness) and by portuguese funds through FCT (Foundation for Science and Technology) within the projects FCOMP-01-0124-FEDER-022674 and PEst-C/MAT/UI0013/2011.

References

1. Aguirre, A.H., Rionda, S.B., Coello Coello, C.A., Lizrraga, G.L., Montes, E.M.: Handling constraints using multiobjective optimization concepts. International Journal for Numerical Methods in Engineering 59, 1989-2017 (2004).
2. Akay, B., Karaboga, D.: Artificial bee colony algorithm for large-scale problems and engineering design optimization, Journal of Intelligence Manufacturing, 23(4), 1001–1014 (2012).
3. Ali M.M., Golalikhani M.: An electromagnetism-like method for nonlinearly constrained global optimization. Computers and Mathematics with Applications, 60, 2279–2285 (2010).
4. Ali, M.M., Kajee-Bagdadi, Z.: A local exploration-based differential evolution algorithm for constrained global optimization, Applied Mathematics and Computation, 208(1), 31–48 (2009).
5. Ali, M.M., Zhu, W.X.: A penalty function-based differential evolution algorithm for constrained global optimization. Computational Optimization and Applications, DOI 10.1007/s10589-012-9498-3 (2012).
6. Audet, C., Dennis, Jr., J.E.: A pattern search filter method for nonlinear programming without derivatives, SIAM Journal on Optimization, 14(4), 980–1010 (2004).
7. Barbosa, H.J.C., Lemonge, A.C.C.: An adaptive penalty method for genetic algorithms in constrained optimization problems, in Frontiers in Evolutionary Robotics, H. Iba (ed.) 34 pages, (ISBN: 978-3-902613-19-6) I-Tech Education Publ., Austria (2008).
8. Bertsekas, D.P.: Nonlinear Programming, 2nd edn. Athena Scientific, Belmont (1999).
9. Birgin, E.G., Floudas, C.A., Martínez, J.M.: Global minimization using an Augmented Lagrangian method with variable lower-level constraints, Technical Report MCDO121206, January 22, 2007.
10. Birgin, E.G., Floudas, C.A., Martínez, J.M.: Global minimization using an Augmented Lagrangian method with variable lower-level constraints, Mathematical Programming, Ser. A, 125(1), 139–162 (2010).
11. Birgin, E.G., Martínez, J.M.: Augmented Lagrangian method with nonmonotone penalty parameters for constrained optimization, Computational Optimization and Applications, 51(3), 941–965 (2012).
12. Chootinan, P., Chen, A.: Constrained handling in genetic algorithms using a gradient-based repair method. Computers and Operations Research 33, 2263–2281 (2006).
13. Civicioglu, P., Besdok, E.: A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms. Artificial Intelligence Review DOI 10.1007/s10462-011-9276-0 (2012).

14. Coello Coello C.A.: Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Computer Methods in Applied Mechanics and Engineering, 191(11), 1245–1287 (2002).
15. Costa, L., Espírito Santo, I.A.C.P., Fernandes, E.M.G.P.: A hybrid genetic pattern search augmented Lagrangian method for constrained global optimization, Applied Mathematics and Computation, 218(18), 9415–9426 (2012).
16. Costa, M.F.P., Fernandes, E.M.G.P.: Assessing the potential of interior point barrier filter line search methods: nonmonotone versus monotone approach, Optimization, 60(10-11), 1251–1268 (2011).
17. Deb, K., Srivastava, S.: A genetic algorithm based augmented Lagrangian method for constrained optimization. Computational Optimization and Applications, 53(3), 869–902 (2012).
18. Dennis Jr., J.E., Price, C.J., Coope, I.D.: Direct search methods for nonlinear constrained optimization using filters and frames, Optimization and Engineering, 5, 123–144 (2004).
19. Di Pillo, G., Lucidi, S., Rinaldi, F.: An approach to constrained global optimization based on exact penalty functions, Journal of Global Optimization, 54, 251–260 (2012).
20. Durrett, R.: Probability: Theory and Examples, Vol. 3, 4th. edn., Cambridge University Press (2010).
21. Fletcher, R., Leyffer, S.: Nonlinear programming without a penalty function, Mathematical Programming, 91, 239–269 (2002).
22. Fletcher, R., Leyffer, S., Toint, Ph.L.: A brief history of filter methods, SIAG/Optimization Views-and-News, 18(1), 2–12 (2007).
23. Gould, N.I.M., Leyffer, S., Toint, Ph.L.: A multidimensional filter algorithm for nonlinear equations and nonlinear least squares, SIAM Journal on Optimization, 15, 17–38 (2004).
24. Gould, N.I.M., Toint Ph.L.: Global convergence of a non-monotone trust-region filter algorithm for nonlinear programming. in Hager, W.W., Huang, S.-J., Pardalos, P.M., Prokopyev, O.A. (eds.), Multiscale Optimization Methods and Applications, Springer 2006.
25. Gómez, W., Remírez, H.: A filter algorithm for nonlinear semidefinite programming, Computational & Applied Mathematics, 29(2), 297–328 (2010).
26. Gonzaga, C.C., Castillo, R.A.: A nonlinear programming algorithm based on non-coercive penalty functions, Mathematical Programming, Ser. A, 96, 87–101 (2003).
27. Greenwood, G.W., Shu Q.J.: Convergence in evolutionary programs with self-adaptation, Evolutionary Computation, 9(2), 147–157 (2001).
28. He, Q., Wang, L.: A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization, Applied Mathematics and Computation, 186, 1407–1422 (2007).
29. Hedar, A.-R., Fahim, A.: Filter-based genetic algorithm for mixed variable programming, Numerical Algebra, Control and Optimization, 1(1), 99–116 (2011).
30. Hedar, A.-R., Fukushima, M.: Derivative-free filter simulated annealing method for constrained continuous global optimization, Journal of Global Optimization, 35(4), 521–549 (2006).
31. Hendrix, E.M.T., G.-Tóth, B.: Introduction to Nonlinear and Global Optimization, Optimization and its Applications 37, Springer-Verlag (2010).
32. Jansen, P.W., Perez, R.E.: Constrained structural design optimization via a parallel augmented Lagrangian particle swarm optimization approach, Computers and Structures, 89, 1352–1366 (2011).
33. Jiang, M., Wang, Y., Pfletschinger, S., Lagunas, M.A., Yuan, D.: Optimal multiuser detection with artificial fish swarm algorithm, CCIS 2, ICIC 2007, D.-S. Huang et al. (eds.) 1084–1093 Springer-Verlag (2007).
34. Jin, Z., Wang, Y.: An improved line search filter method for the system of nonlinear equations, Journal of Applied Mathematics, 2012, ID 273141, 15 pages (2012).
35. Karas, E.W., Gonzaga, C.C., Ribeiro, A.A.: Local convergence of filter methods for equality constrained non-linear programming, Optimization, 59(8), 1153–1171 (2010).
36. Karas, E., Ribeiro, A., Sagastizábal, C., Solodov, M.: A bundle-filter method for nonsmooth convex constrained optimization, Mathematical Programming Ser. B, 116, 297–320 (2009).
37. Karr, A.F.: Probability, Springer Texts in Statistics, Springer-Verlag, New York (1993).
38. Lewis, R.M., Torczon, V.: A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds, SIAM Journal on Optimization, 12, 1075–1089 (2002).
39. Liu, J.-L., Lin, J.-H.: Evolutionary computation of unconstrained and constrained problems using a novel momentum-type particle swarm optimization, Engineering Optimization, 39(3), 287–305 (2007).
40. Luo, H., Sun, X., Wu, H.: Convergence properties of augmented Lagrangian methods for constrained global optimization, Optimization Methods and Software, 23(5), 763–778 (2008).
41. Mahdavi, M., Fesanghary, M., Damangir, E.: An improved search algorithm for solving optimization problems, Applied Mathematics and Computation, 188, 1567–1579 (2007).
42. Neshat, M., Sepidnam, G., Sargolzaei, M., Toosi, A.N.: Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications, Artificial Intelligence Review, DOI:10.1007/s10462-012-9342-2 (2012).

43. Nie, P.-y., Fan, J.-y.: A derivative-free filter method for solving nonlinear complementarity problems, Applied Mathematics and Computation, 161, 787–797 (2005).
44. Nie, P.-y., Lai, M.-y., Zhu, S.-j., Zhang, P.a.: A line search filter approach for the system of nonlinear equations, Computers and Mathematics with Applications, 55(9), 2134–2141 (2008).
45. Pereira, A.I., Costa, M.F.P., Fernandes, E.M.G.P.: Interior point filter method for semi-infinite programming problems, Optimization, 60(10-11) 1309–1338 (2011).
46. Petalas, Y.G., Parsopoulos, K.E., Vrahatis, M.N.: Memetic particle swarm optimization, Annals of Operations Research, 156, 99–127 (2007).
47. Ribeiro, A.A., Karas, E.W., Gonzaga, C.C.: Global convergence of filter methods for nonlinear programming, SIAM journal on Optimization, 19(3), 1231–1249 (2008).
48. Rocha, A.M.A.C., Fernandes, E.M.G.P.: Numerical study of augmented Lagrangian algorithms for constrained global optimization. Optimization, 60(10-11), 1359–1378 (2011).
49. Rocha, A.M.A.C., Costa, M.F.P., Fernandes, E.M.G.P.: An artificial fish swarm filter-based method for constrained global optimization, Lecture Notes in Computer Science, Vol. 7335, ICCSA 2012 Part III, Murgante B., Gervasi O., Misra S., Nedjah N., Rocha A.M.A.C., Taniar D., Apduhan B.O. (eds.) 57–71 (2012).
50. Rocha, A.M.A.C., Fernandes, E.M.G.P., Martins, T.F.M.C.: Novel fish swarm heuristics for bound constrained global optimzation problems, Lecture Notes in Computer Science, Vol. 6784, ICCSA 2011 Part III, Murgante B., Gervasi O., Iglesias A., Taniar D., Apduhan B. (eds.) 185–199 (2011).
51. Rocha, A.M.A.C., Martins, T.F.M.C., Fernandes, E.M.G.P.: An augmented Lagrangian fish swarm based method for global optimization, Journal of Computational and Applied Mathematics, 235(16), 4611–4620 (2011).
52. Scholz, D.: Geometric branch-and-bound methods for constrained global optimization problems, Journal of Global Optimization, DOI 10.1007/s10898-012-9961-9 (2012).
53. Sedlaczek, K., Eberhard, P.: Augmented Lagrangian particle swarm optimization in mechanism design, Journal of System Design and Dynamics, 1, 410–421 (2007).
54. Shen, C., Leyffer, S., Fletcher, R.: A nonmonotone filter method for nonlinear optimization, Computational Optimization and Applications, 52(3), 583–607 (2012).
55. Silva E.K., Barbosa H.J.C., Lemonge A.C.C.: An adaptive constraint handling technique for differential evolution with dynamic use of variants in engineering optimization, Optimization and Engineering, 12, 31–54 (2011).
56. Silva, R., Ulbrich, M., Ulbrich, S., Vicente, L.N.: A globally convergent primal-dual interior-point filter method for nonlinear programming: new filter optimality measures and computational results. Technical Report 08-49,, Department of Mathematics, University of Coimbra, 2009.
57. Tahk, M.-J., Woo, H.-W., Park, M.-S.: A hybrid optimization method of evolutionary and gradient search, Engineering Optimization, 39, 87–104 (2007).
58. Tsoulos, I.G.: Solving constrained optimization problems using a novel genetic algorithm, Applied Mathematics and Computation, 208, 273–283 (2009).
59. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Mathematical Programming, 106, 25–57 (2006).
60. Wang, H., Pu, D.: A nonmonotone filter trust region method for the system of nonlinear equations, Applied Mathematical Modelling, 37(1-2), 498–506 (2013).
61. Wang, X., Gao, N., Cai, S., Huang, M.: An artificial fish swarm algorithm based and ABC supported QoS unicast routing scheme in NGI, Lecture Notes in Computer Science, ISPA, G. Min et al. (eds.) 4331, 205–214 (2006).
62. Wu, T., Sun, L.: A filter-based pattern search method for unconstrained optimization, Numerical Mathematics, 15(3), 209–216 (2006).
63. Yu, K., Pu, D.: A nonmonotone filter trust region method for nonlinear constrained optimization, Journal of Computational and Applied Mathematics, 223(1), 230–239 (2009).
64. Zahara, E., Hu, C.-H.: Solving constrained optimization problems with hybrid particle swarm optimization, Engineering Optimization, 40, 1031–1049 (2008).
65. Zhao, Q., Guo, N.: A nonmonotone filter method for minimax problems, Applied Mathematics, 2, 1372–1377 (2011).
66. Zhou, Y.Y., Yang, X.Q.: Augmented Lagrangian functions for constrained optimization problems, Journal of Global Optimization, 52(1), 95–108 (2012).