J Glob Optim
DOI 10.1007/s10898-015-0388-y

Stochastic filter methods for generally constrained global optimization

C. J. Price[1] · M. Reale[1] · B. L. Robertson[1]

Received: 25 November 2014 / Accepted: 15 November 2015

Abstract A filter based template for bound and otherwise constrained global optimization of non-smooth black-box functions is presented. The constraints must include finite upper and lower bounds, and can include nonlinear equality and inequality constraints. Almost sure convergence is shown for a wide class of algorithms conforming to this template. An existing method for bound constrained global optimization (OSCARS) is easily modified to conform to this template. Numerical results show the modified OSCARS is competitive with other methods on test problems including those listed by Koziel and Michalewicz.

Keywords Filter · ARS · OSCARS · Direct search · Bound and otherwise constrained global optimization

1 Introduction

The bound and otherwise constrained global optimization problem is

$$\min_{x\in\Omega} f(x) \quad \text{subject to} \quad g(x) \le 0 \tag{1}$$

where f is the objective function. The constraint function g is a vector function mapping $\mathbb{R}^n \to \mathbb{R}^q$, and $\Omega \subset \mathbb{R}^n$ is a finite box of the form

$$\Omega = \left\{x \in \mathbb{R}^n : L_i \le x_i \le U_i \quad \forall i = 1, 2, \ldots, n\right\}.$$

✉ C. J. Price
C.Price@math.canterbury.ac.nz

M. Reale
marco.reale@canterbury.ac.nz

B. L. Robertson
blair.robertson@canterbury.ac.nz

[1] Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand

Published online: 21 November 2015

Here x_i is the ith element of x, and all upper and lower bounds U_i and L_i are finite, with $L_i < U_i$ for all i. For simplicity only general inequality constraints are included in (1). Equality constraints can be included as a pair of inequality constraints, without loss of generality.

We assume that f and g are black-box functions which, when supplied with a point x, return $f(x)$ and $g(x)$. It is also assumed that f and $[g]_+$ are Lebesgue measurable, where $[g(x)]_+ \in \mathbb{R}^q$ takes the element-wise maximum of $g(x)$ and zero. Finally we assume f is bounded below, and that f and $[g]_+$ are lower semi-continuous at points of interest.

There are a number of different strategies to address (1), some of which reduce (1) to minimizing an appropriate penalty function over Ω [5]. A related approach is [9], which applies the DIRECT method of Jones et al. [10] to an auxiliary function. Interval methods [13,20] have also been used to solve (1) via a branch and bound strategy. Other approaches include the use of approximations to the objective function [16], and filled functions [23]. In addition there are many stochastic methods including fish swarm [18], particle swarm [21], simulated annealing [8], evolutionary algorithms [19], and electromagnetism-like methods [1].

Many of the methods above make use of information not available in non-smooth black-box optimization, such as differentiability of f and g [5,22–24], or the functions' analytic forms [13,20]. In contrast methods such as [8,9,18,19,21] require only the functions' values at each iterate. Since these methods neither use nor require additional information or properties about f and g, they can be easily applied to non-smooth black-box functions, as is the intent of this paper. Allowing non-linear equality constraints means feasible point algorithms are not an option. Dealing with infeasible points presents two conflicting aims: attaining feasibility and minimizing f. One way to reconcile these two aims is to use an exact penalty function, which requires an appropriate choice of penalty parameter(s). Di Pillo et al. [5] examine this when f and g are continuously differentiable. Since (1) might not be smooth, picking such parameters is usually hard. One could work with a variety of different penalty parameter values simultaneously, but, depending on the relative scaling of f and g, the best such values could be arbitrarily large or small. Instead, like [8], we opt for a filter.

In general terms, global optimization algorithms calculate f and g at a sequence of sample points in Ω. Subsequent sample points are selected based on information from earlier sample points. For example, a method could explore more thoroughly the vicinity of promising sample points than other parts of Ω. One way to identify promising sample points is by their objective function values and constraint violations. If we retain only points which are clearly better than any other sample point in either objective function value or constraint violation or both, then the points we retain form a filter. Discarding poor sample points in this fashion can lower computational overheads and storage requirements as well as improve sampling effectiveness.

In this paper we formulate a general framework for solving (1) using filters [6]. The basic approach of the filter template is to generate an infinite sequence of iterates $\{x^{(k)}\}_{k=0}^{\infty}$ in Ω. An infinite subset of these iterates are drawn randomly from Ω in order to guarantee convergence almost surely. Others are drawn from subregions of Ω with the intent of improving the convergence rate. The existence of a sequence of filter points converging to solution(s) of (1) with probability one can be shown under mild conditions. A modified version of one side cut accelerated random search (hereafter OSCARS) [15] conforming to the filter template is also tested.

Accelerated random search (ARS) [3], and its offshoot OSCARS [15], are methods for bound constrained global optimization which use the objective function *only* to determine which of a pair of points is better than the other. Hence they can easily be modified to address other problems such as (1) when an alternative way of doing pairwise comparisons (e.g. via filters) exists.

ARS searches about the current best known point (hereafter the control point) by polling randomly in a finite sequence of nested boxes until a better point is found. This better point then becomes the next control point, and the process is repeated. If the sequence of nested boxes is exhausted without improvement, ARS continues polling cyclically through the sequence of boxes. The first box in the sequence is Ω. Each subsequent box is contained in its predecessor, and is the intersection of a scaled version of Ω centred on the current iterate with Ω itself.

The principal difference between ARS and OSCARS is the latter does not shrink the polling box in every direction after a failed poll. Using c as the control point, and $x^{(k)}$ as the k^{th} random polling point, OSCARS shortens the box along one coordinate axis so that c remains in the shortened box, but $x^{(k)}$ does not. The axis used is the one along which $x^{(k)}$ and c have the greatest separation. Like ARS, this process continues until either a better point than c is found, or the polling box becomes too small, after which the polling box is reset to Ω. If a better point has been found, then c is set to this better point.

This paper is organized as follows. The next section formally defines what is meant by a solution to (1). Section 3 describes the stochastic filter template, and establishes convergence almost surely. Sections 4 and 5 describe and numerically test a filter version of OSCARS which conforms to the template. Concluding remarks appear in Sect. 6.

2 Theoretical development

The feasible region Φ is

$$\Phi = \{x \in \Omega : g(x) \leq 0\}.$$

The total constraint violation at a point $x \in \Omega$ is measured using

$$\theta(x) = G\left(\|[g(x)]_+\|\right) \tag{2}$$

where $[g(x)]_+ \in \mathbb{R}^q$ takes the element-wise maximum of $g(x)$ and zero, and G is a continuous strictly increasing function satisfying $G(0) = 0$. The inclusion of the function G allows such choices as

$$\theta(x) = \|[g(x)]_+\|_2 + \|[g(x)]_+\|_2^2 \tag{3}$$

which is used to generate the numerical results presented herein. Continuity of G ensures points with small values for $[g]_+$ have correspondingly small values for θ. The use of weighted norms in (2) permits constraint scaling to be included.

Our approach is intended to cope with ‘black-box’ objective and constraint functions. The convergence analysis requires that the objective function f be Lebesgue measurable, bounded below, and lower semi-continuous at points of interest. The minimum conditions on g are that θ is both Lebesgue measurable, and lower semi-continuous. This implies the feasible region Φ is a closed set under the standard topology on $\mathbb{R}^n$ restricted to Ω. This topology is used exclusively in this paper.

First we define precisely what is regarded as an acceptable solution to (1). This definition makes use of infeasible points, and so we define the set of points $\Phi(\eta)$ in which the constraint violation is at most η as follows.

$$\Phi(\eta) = \{x \in \Omega : \theta(x) \leq \eta\}.$$

Lower semi-continuity of the measure of infeasibility θ means each $\Phi(\eta)$ is a closed set. Since Ω is bounded, each $\Phi(\eta)$ is compact. In particular, the feasible region $\Phi = \Phi(0)$ is compact.

Our intent is to explore methods which solve (1) by random search over Ω and various subregions of Ω. Convergence is guaranteed almost surely through sample points (or iterates) chosen randomly from Ω. Hence sets of Lebesgue measure zero are essentially invisible because sampling them is a probability zero event. This motivates the following definitions. Let m_L be the Lebesgue measure.

Definition 1 (*Essential global minimum of f over $\Phi(\eta)$*) Let $\gamma(\eta)$ be the infimum of infinity and all values f_{test} for which

$$m_L\left(\{z \in \Phi(\eta) : f(z) < f_{\text{test}}\}\right) > 0. \tag{4}$$

Then $\gamma(\eta)$ is the essential global minimum of f over $\Phi(\eta)$. Any point $x \in \Phi(\eta)$ satisfying $f(x) \le \gamma(\eta)$ is called an essential global minimizer of f over $\Phi(\eta)$.

When $\eta = 0$, $\gamma(0)$ gives the essential global minimum of f over the feasible region Φ. The definition of $\gamma(0)$ only considers feasible points. Allowing nearly feasible points to be considered gives:

Definition 2 (*Essential global solution of* (1)) Let f^*_{egs} be the infimum of infinity and all values f_{test} which satisfy

$$m_L\left(\{z \in \Phi(\epsilon) : f(z) < f_{\text{test}}\}\right) > 0, \quad \forall \epsilon > 0. \tag{5}$$

Then f^*_{egs} is the essential global solution value of (1). Any feasible point x satisfying $f(x) \le f^*_{\text{egs}}$ is called an essential global solution point of (1).

The set of all essential global solution points is denoted S^*. Given that we are solving (1) by generating random sample points in Ω and various subregions of Ω, attaining the essential global solution value is the best we can realistically aim for.

Clearly $\gamma(\eta) \le f^*_{\text{egs}}$ for all positive η. This follows directly from replacing "$\forall\epsilon > 0$" with "$\epsilon = \eta$" in (5). In fact

$$\sup\{\gamma(\eta) : \eta > 0\} = \lim_{\eta \downarrow 0} \gamma(\eta) = f^*_{\text{egs}} \tag{6}$$

where the limit exists and equals the supremum because γ is a decreasing function of η. Clearly f^*_{egs} is an upper bound on this limit. Now, for any f_{test}, f_{test} satisfies (5) iff f_{test} satisfies (4) for all $\eta > 0$. The right hand equality in (6) follows because f^*_{egs} is the infimum of all f_{test} values satisfying (5). Note that putting $\eta = 0$ in $\gamma(\eta)$ does not yield f^*_{egs}. The latter's definition considers points which do not lie in Φ, whereas the former (with $\eta = 0$) does not. The difference between these quantities is easily seen from the following example.

Example 1

$$f = x_1^2 + (x_2 - 2)^2 \quad g = (x_1 - x_2)^2 \max(x_1, x_2), \quad \text{and} \quad \Omega = [-2, 2]^2.$$

The feasible region consists of the union of bottom left quadrant of Ω and the line $x_1 = x_2$ with $0 \le x_1 \le 2$. The global minimum and essential global solution f^*_{egs} both equal 2, and occur at (1, 1), as is shown in Fig. 1. In contrast $\gamma(0) = 4$ and this value is achieved at the origin. If Ω is altered to $\Omega = [0, 2]^2$ then the global minimizer and essential global solution are unchanged, but $\gamma(0) = \infty$.

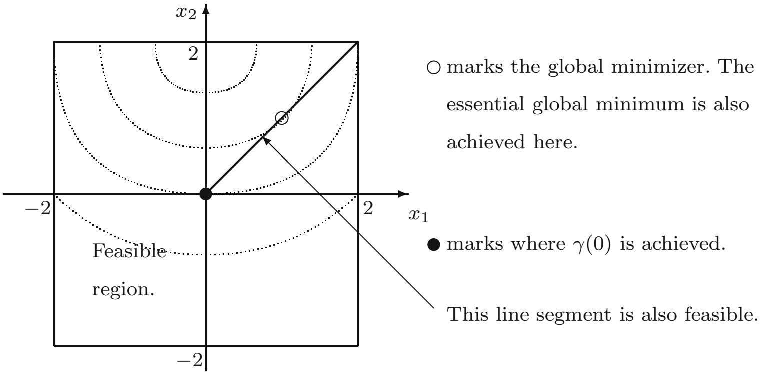
x_2
2
-2
2
x_1
-2
Feasible
region.
○ marks the global minimizer. The essential global minimum is also achieved here.
● marks where $\gamma(0)$ is achieved.
This line segment is also feasible.

Fig. 1 The feasible region (*boundary in bold*) and level curves of f (*dotted*) for Example 1

3 Filters and convergence

The notion of a filter [6] is based on a two dimensional 'ordering' of points in Ω in terms of their f and θ values. If one point (x say) has lower values for both f and θ than another (say y), then x is clearly a better point than y. Similarly x is clearly worse if both its values are higher; and x is comparable to y if one value is higher, and the other lower. At highly infeasible points objective function values are often uninformative, and f might not even be defined in such regions. For such points one point is better than another only if it has lower infeasibility. To be precise, a point x weakly dominates a second point y (written $x \preceq y$) if and only if either

a. both $f(x) \leq f(y)$ and $\theta(x) \leq \theta(y)$; or
b. both $\theta(y) > \Theta$ and $\theta(x) \leq \theta(y)$.

Here Θ is the θ value above which points which are regarded as highly infeasible. If either both inequalities in (a) hold and at least one is strict, or both inequalities in (b) hold and the second one is strict, then we say x dominates y (written $x \prec y$).

A filter F is simply a set of sample points with the property that no point in the filter is dominated by any other point in the filter [6]. In contrast to [6], feasible points are included directly in our filter, and do not have any special status there. A filter can be used to divide Ω into three subsets according to the values taken by f and θ. The first of these is the set of 'worse' points. This set contains all points in Ω which are dominated by at least one point in the filter. The second set contains all points which dominate at least one filter point. This is the set of 'better' points. All remaining points neither dominate, nor are dominated by, any point in the filter F. These points are called comparable points to the filter F.

Filters may be used to solve (1) when the sequence of iterates $\{x^{(k)}\}_{k=0}^{\infty}$ contains an infinite subsequence of samples drawn randomly from Ω. At iteration k the filter $F_k \subseteq \{x^{(0)}, \ldots, x^{(k)}\}$ is formed by updating the previous iteration's filter with the current iterate $x^{(k)}$. If $x^{(k)} \prec z$ for some $z \in F_{k-1}$ then F_k is set equal to $F_{k-1} \cup \{x^{(k)}\}$. All points in F_k which are dominated by $x^{(k)}$ are then removed from F_k. Alternatively if $z \prec x^{(k)}$ for some $z \in F_{k-1}$, then F_k is set equal to F_{k-1}. The remaining case is when $x^{(k)}$ is comparable to F_{k-1}. Following [6] we include all comparable points in the filter.

The tactic of including all unfiltered (i.e. comparable and better points) in the filter can lead to the size $|F_k|$ of F_k increasing steadily as $k \to \infty$. Whilst this is not desirable, it is not uncommon for global optimization methods to have storage requirements which increase steadily as k increases. For example [2,10,14,17] store and process every iterate generated, or at least a fixed percentage of them. Clearly filters usually discard many iterates, and so this approach is not unreasonable. Nevertheless strategies which limit the number of points $|F|$ in F might be useful. We examine one such pruning process, and then describe the filter template.

When the size $|F|$ of F becomes excessive, the filter points are grouped into bands, where points in the same band have similar θ values. Pruning retains only the points which have the least f value in some band, with one exception. The point which is the least infeasible infeasible point w in F is also retained, even when F contains a feasible point. Without automatically retaining w there is a risk of failing to reduce infeasibility by cyclically generating points such as w with new low values of θ, and then deleting them during pruning because of another filter point z in the same band with $f(z) < f(w)$ and $\theta(z) > \theta(w)$.

A set of pruning marks $\mathcal{M} \subseteq [0, \infty)$ is chosen, where

$$0 \in \mathcal{M} \quad \text{and} \quad \inf\{\mu \in \mathcal{M} : \mu > 0\} = 0. \tag{7}$$

A filter F can be pruned using these marks to get a filter P as follows. The least infeasible infeasible point is always included in P. The remaining points in P are those which minimize f over $F \cap \Phi(\mu)$ for some $\mu \in \mathcal{M}$ satisfying $\mu \le \Theta$. The condition $0 \in \mathcal{M}$ ensures that if any feasible points are known, the one with the least f value is included in the pruned filter. The second condition in (7) means $\mathcal{M}$ contains an infinite sequence of marks converging to zero. Hence if F contains a sequence of infeasible iterates converging to a feasible point, then P retains a subsequence with the same property. There is no requirement that $\mathcal{M}$ contain members which are arbitrarily large.

A sensible choice of $\mathcal{M}$ could be, for example,

$$\mathcal{M} = \{0\} \cup \{\beta^j \tau : \beta^j \tau \le \Theta \text{ and } j \in \mathbb{Z}\} \tag{8}$$

where $\mathbb{Z}$ is the integers, $\tau > 0$, and $\beta > 1$. The use of a limit Θ above which points are judged purely on infeasibility means pruning marks greater than Θ are superfluous.

A template for a filter based method is stated as Algorithm 1. The first step initializes the filter. Steps 2 to 5 form a loop which processes one point in each iteration. The iterate $x^{(k)}$ is generated in step 2. The cases when $x^{(k)}$ is a worse, better, or comparable point to the filter F_{k-1} are handled in steps 3, 4, and 5 respectively. Step 5 permits pruning of the filter, but does not require it. Provided $\mathcal{M}$ satisfies (7), a method conforming to this filter template will generate a sequence of iterates converging to a solution of (1) almost surely when the stopping conditions are suspended. Furthermore, with or without pruning, the sequence of filters will retain this sequence of iterates. Pruning can reduce the mark Θ above which points are judged solely on their θ values, however the sequence of Θ values must remain bounded away from zero.

Algorithm 1 *The filter based template for generally constrained optimization.*

```
1. Choose $x^{(0)} \in \Omega$. Set $F_0 = \{x^{(0)}\}$, set $k = 0$, and choose $\Theta \le \Theta_{\max}$.
2. Increment $k$. If stopping conditions hold exit the algorithm. Otherwise choose $x^{(k)} \in \Omega$.
3. If $z \prec x^{(k)}$ for some $z \in F_{k-1}$, set $F_k = F_{k-1}$ and go to step 2.
4. Set $F_{\text{temp}} = \{x^{(k)}\} \cup F_{k-1}$ and eliminate all dominated points in $F_{\text{temp}}$. If $x^{(k)} \prec z$
   for at least one $z \in F_{k-1}$, set $F_k = F_{\text{temp}}$ and go to step 2.
5. If desired, prune $F_{\text{temp}}$ to get $F_k$, and reduce $\Theta$ if necessary. Otherwise set $F_k =$
   $F_{\text{temp}}$. Go to step 2.
```

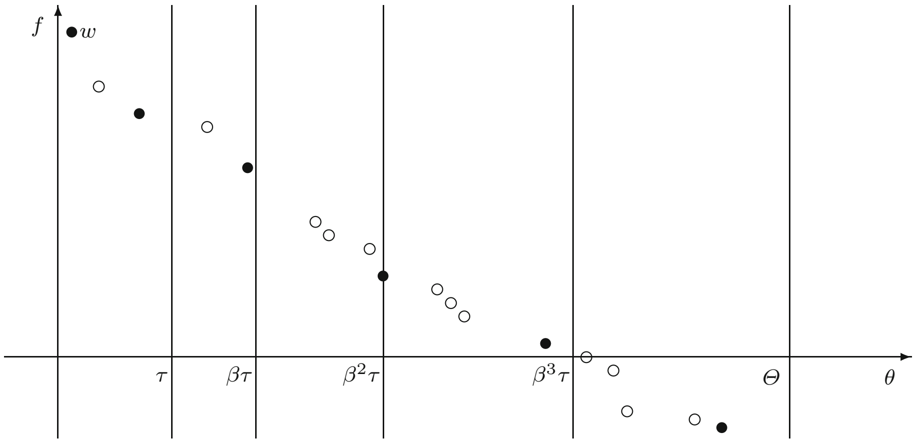

Fig. 2 This figure shows points in a filter before (*circles and dots*), and after pruning (*dots only*). The filter does not contain a feasible point. *Pruning marks*, as given in Eq. (9), are shown as *vertical lines* at $\mu = \beta^j\tau$ where τ is the tolerance on constraint violations, and β is the scale factor between adjacent marks. Here $J_{\min} = 0$ so the lowest positive mark is $\mu = \tau$. The marks do not extend beyond Θ. The least infeasible infeasible point, marked w, is always included in the pruned filter. The remaining pruned filter points each minimize f over the set of filter points with constraint violations θ less than or equal to some *pruning mark*

Pruning, as so far described, still allows the size of the filter to become arbitrarily large. However for a practical algorithm, there is a tolerance $\tau > 0$ on constraint violation. Below this tolerance points are considered "feasible for all practical purposes" and such points are judged only on their objective function value. Moreover the filter has an upper limit Θ above which objective function values are ignored. For appropriate $\mathcal{M}$, these two limits yields a finite upper bound on the size of a pruned filter. Instead of defining the maximum filter size N in terms of Θ, we define Θ and $\mathcal{M}$ in terms of N, τ, β, and a minimum value $J_{\min} \leq 0$ on the index j, as follows:

$$\mathcal{M} = \{0\} \cup \{\beta^j\tau : \beta^j\tau \leq \Theta \text{ and } j \in [J_{\min}, \infty) \cap \mathbb{Z}\}. \tag{9}$$

The algorithm starts with a far larger value for Θ (such as $\Theta = \infty$) and reduces it as necessary to enable pruning to limit the filter size to N. This allows points with very large θ values to be in the filter during early iterations, which is a desirable feature. As points with low θ values are found, the limit Θ is steadily reduced to keep $|F| \leq N$, thus directing attention towards feasible parts of Ω as they are identified. The value $J_{\min}$ allows points to be retained which have constraint violations less than the stopping tolerance τ, which might be useful in reducing f. An illustration of a filter, before and after pruning, is given in Fig. 2.

3.1 Convergence

In this and the following subsection, almost sure convergence of methods conforming to the template is shown, subject to the following assumption holding.

Assumption 3 (a) The objective function f is lower semi-continuous (and hence bounded below) on $\Phi(\Theta_{\max})$.

(b) The constraint violation θ is lower semi-continuous on Ω.

(c) The set of pruning marks $\mathcal{M}$ satisfies (7).

(d) The sequence of iterates $\{x^{(k)}\}_{k=0}^{\infty}$ almost surely contains an infinite number of sample points drawn randomly from Ω.

Parts (a) and (b) of this assumption are conditions on the problem. Since Θ is never increased, it suffices that part (a) holds for the initial value of Θ. Parts (c) and (d) are a conditions on any algorithm conforming to the template. Condition (d) requires that the algorithm will, almost surely, look everywhere eventually. Conditions like this are very common in global optimization when confronted with black-box functions. In contrast, if global information such as Lipschitz constant is available, or interval arithmetic methods can be employed, the need to 'look everywhere' might be superfluous.

The analysis proceeds under the assumption that stopping conditions are removed and an algorithm conforming to the template is run forever. Under these conditions a variety of convergence results are shown to hold almost surely. The results in this, and the following subsection address the cases when f^*_{egs} is infinite, and finite respectively. The first of these looks at the ability to attain feasibility from the infeasible region.

Theorem 4 *Assume all iterates are infeasible, and let w_k minimize θ over F_k for each $k \in \mathbb{N}$. Then every cluster point w^* of the sequence of least infeasible infeasible filter points $\{w_k\}$ is an essential global minimizer of θ over Ω, almost surely.*

Proof The pruning strategy always retains the infeasible point with the least θ value. Hence w_k persists in each filter until an infeasible point w is found satisfying either $\theta(w) < \theta(w_k)$, or both $\theta(w) = \theta(w_k)$ and $f(w) < f(w_k)$. Thus $\{\theta(w_k)\}$ is a non-strictly decreasing sequence. This, and Assumption 3d imply

$$\lim_{k\to\infty} \theta(w_k) \le \inf\{\theta_{\text{test}} \in \mathbb{R} : m_L(\Phi(\theta_{\text{test}})) > 0\} \quad \text{almost surely.}$$

Lower semi-continuity of θ implies any cluster point w^* of $\{w_k\}$ is an essential global minimizer of θ over Ω, almost surely. □

This shows the template will asymptotically find a feasible point provided the measure of $\Phi(\eta)$ is positive for all $\eta > 0$. This latter condition holds when, for example, the feasible region has positive measure, or a feasible point exists at which g is continuous.

The next result applies directly to the sequence of best known feasible iterates, and shows that its cluster points are essential global minimizers of f over $\Phi(0)$. A more general result is shown first, with this as a corollary. We define the best known point $b_k(\mu)$ as the point which minimizes f over $F_k \cap \Phi(\mu)$, where $\mu \in \mathcal{M}$.

Theorem 5 *Let f and θ be lower semi-continuous on $\Phi(\mu)$, where $\mu \in \mathcal{M}$ satisfies $m_L(\Phi(\mu)) > 0$. Then cluster points of the sequence of best known points $\{b_k(\mu)\} \subset \Phi(\mu)$ are essential global minimizers of f over $\Phi(\mu)$, with probability one.*

Proof Let b^* be a cluster point of $\{b_k(\mu)\}$ which is not an essential global minimizer of f over $\Phi(\mu)$ in the topology induced on Ω by the standard topology for $\mathbb{R}^n$. Lower semi-continuity of θ means $\Phi(\mu)$ is closed. Hence $b^* \in \Phi(\mu)$. Now $\{x \in \Phi(\mu) : f(x) < f(b^*)\}$ has positive Lebesgue measure, so this set is sampled almost surely, by Assumption 3d. Hence a $b_k(\mu)$ exists satisfying $f(b_k(\mu)) < f(b^*)$, almost surely. Now $\{f(b_k(\mu))\}$ is a decreasing sequence, so lower semi-continuity of f means b^* can not be a cluster point of $\{b_k(\mu)\}$, almost surely. □

For any $\mu \in \mathcal{M}$, this result shows almost sure convergence to a point which achieves the essential global minimum $\gamma(\mu)$ over $\Phi(\mu)$. In the case when $\mu = 0$ it yields the following corollary.

Corollary 6 *Let f be lower semi-continuous on $\Phi \equiv \Phi(0)$ and let $m_L(\Phi(0)) > 0$. Then the cluster points of the sequence of best known feasible points $\{b_k(0)\} \subset \Phi$ are essential global minimizers of f over Φ, almost surely.*

Proof The result follows directly from Theorem 5 on putting $\mu = 0$. □

3.2 Convergence to an essential global solution

Finally we establish almost sure convergence to a point in the essential global solution set S^* for (1), when f^*_{egs} is finite. We note that this condition implies $m_L(\Phi(\eta)) > 0$ for all $\eta > 0$ via Definition 2. We define open neighbourhoods of S^* of the form

$$S^*_\delta = \left\{x \in \Omega : d(x, S^*) < \delta\right\} \quad \text{for each} \quad \delta > 0,$$

where $d(x, S^*)$ is the distance between the point x and the optimal set S^*; that is to say

$$d(x, S^*) = \inf\{\|x - s\|_2 : s \in S^*\}.$$

The next theorem shows that if a point is nearly feasible and nearly optimal for its level of constraint violation, then it is near to an essential global solution of (1). By generating and identifying such points an essential global solution can be located asymptotically.

Theorem 7 *Let f^*_{egs} be finite. For all sufficiently small positive ϵ and $\mu \in \mathcal{M}$,*

$$f(x) < \gamma(\mu) + \epsilon \quad \text{and} \quad x \in \Phi(\mu) \quad \Rightarrow \quad x \in S^*_{\delta(\epsilon,\mu)}$$

where $\delta(\epsilon, \mu) \to 0$ as $\sqrt{\epsilon^2 + \mu^2} \to 0$.

Proof The proof is by contradiction. Assume the theorem is false, and choose monotonically decreasing sequences $\{\epsilon_j\}_{j=1}^\infty$ and $\{\mu_j\}_{j=1}^\infty \subseteq \mathcal{M} - \{0\}$, both converging to zero as $j \to \infty$. Then there exists a positive constant ξ and a sequence $\{y_j\}_{j=1}^\infty$, such that

$$y_j \in \Phi\left(\mu_j\right) \quad \text{and} \quad f(y_j) < \gamma\left(\mu_j\right) + \epsilon_j \quad \forall j \in \mathbb{N} \tag{10}$$

and

$$y_j \notin S^*_\xi \quad \forall j \in \mathbb{N}. \tag{11}$$

The sequence $\{y_j\}$ lies in a compact set, and so must have cluster points. Let one such (arbitrary) cluster point be y_∞. By replacing $\{y_j\}$ with a subsequence of itself if necessary, let $\{y_j\}$ converge to y_∞. Now $\theta(y_j) \to 0$ as $j \to \infty$, and so by the lower semi-continuity of θ, it follows that $\theta(y_\infty) = 0$. Lower semi-continuity of f and (10) yield the left hand inequality in

$$f(y_\infty) \le \lim_{j\to\infty} \gamma\left(\mu_j\right) = f^*_{\text{egs}}.$$

The right hand equality is via Eq. (6). Hence $y_\infty \in S^*$ because $y_\infty \in \Phi$ and $f(y_\infty) \le f^*_{\text{egs}}$. Since $y_j \to y_\infty$ as $j \to \infty$ this contradicts (11). □

This result is directly exploited by the filter method: for any fixed $\epsilon, \mu > 0$ with $\mu \in \mathcal{M}$, any point $x \in \Phi(\mu)$ satisfying $f(x) < \gamma(\mu) + \epsilon$ can only be dominated or pruned by another point satisfying those conditions. Hence once the filter contains a point satisfying these conditions for a specific $\epsilon, \mu > 0$ with $\mu \in \mathcal{M}$, it will always do so. Such points occur almost surely as the algorithm draws an infinite subsequence of sample points randomly from Ω.

Theorem 8 *Let $m_L(\Phi(\eta)) > 0$ for all $\eta > 0$. Then there exists a sequence of filter points $\{z_k\}_{k=1}^{\infty}$ such that any cluster point z^* of $\{z_k\}_{k=1}^{\infty}$ satisfies $z^* \in S^*$ almost surely, where $z_k \in F_k$ for all k.*

Proof Pick arbitrary positive values for $\mu_j \in \mathcal{M} - \{0\}$ and ϵ_j. The definition of $\gamma(\mu_j)$ shows that the set

$$T(\mu_j, \epsilon_j) = \{x \in \Phi(\mu_j): f(x) \leq \gamma(\mu_j) + \epsilon_j\}$$

has strictly positive Lebesgue measure. Hence the probability that a randomly generated sample point in Ω lies in T is strictly positive. The number of sample points randomly drawn from Ω is infinite with probability one, so $\{x^{(k)}\} \cap T$ is non-empty, almost surely. Let z be the first such sample point generated. This point will be placed in the filter in step 4 or 5 of the template. The sample point z can be removed from the pruned filter only if a new point z_+ is found satisfying either $z_+ \prec z$, or both $z_+ \in \Phi(\mu_j)$ and $f(z_+) \leq f(z)$. Both of these possibilities imply $z_+ \in T$.

By considering a sequence of pairs (μ_j, ϵ_j) converging to zero as $j \to \infty$, with $\mu_j \in \mathcal{M}$, it is clear that the sequence of filters contains a sequence of points in $T(\mu_j, \epsilon_j)$ with $(\mu_j, \epsilon_j) \to (0, 0)$, almost surely. Theorem 7 shows that this sequence converges to one or more solutions of (1), as required. □

4 The filter OSCARS algorithm

For convenience we refer to the filter OSCARS method as F- OSCARS, a summary of which is presented as Algorithm 2. In addition to storing z, $f(z)$, and $\theta(z)$ for each filter point $z \in F$, F- OSCARS also stores the lower and upper bounds $\ell_k(z)$ and $u_k(z)$ of $\Omega_k(z)$. The box $\Omega_k(z)$ is the region from which a random sample will be drawn if z is chosen as the control point.

The basic approach of F- OSCARS is as follows: at iteration k the method randomly chooses a control point $c \in F_{k-1}$. The method then draws the next iterate $x^{(k)}$ randomly from $\Omega_k(c)$, calculates f and θ at $x^{(k)}$ and updates the filter according to the template in Algorithm 1. Step 1 in Algorithm 2 initializes the method. Steps 2–5 form a loop which generates and processes one sample point $x^{(k)}$ per iteration. Step 2 generates $x^{(k)}$ and inserts it into the filter. Step 2 then removes any dominated points from the filter. If $x^{(k)}$ is a better point than F_{k-1} no further action needs to be taken, in which case the current iteration ends at step 3. Otherwise, steps 4 and 5 update the filter and other relevant information. Step 5 prunes the filter. When $x^{(k)}$ is a worse point to F_{k-1} we have $F_k = F_{k-1}$ and so no pruning actually occurs, in accordance with the filter template.

In addition to the filter, the box $\Omega_k(c)$ must be updated at iteration k, where c is the control point used at iteration k. This is done using a similar strategy to ARS. The box $\Omega_k(c)$ is shrunk whenever the new sample point $x^{(k)}$ is comparable to or worse than F_{k-1}. The box $\Omega_k(c)$ is shortened using the same mechanism as OSCARS. Using

$$\Omega_k(c) = \{x \in \Omega: \ell_k(c) \leq x \leq u_k(c)\},$$

the current control point c remains in the filter and $\Omega_{k+1}(c) \subset \Omega_k(c)$ is chosen so that c lies in $\Omega_{k+1}(c)$, but $x^{(k)}$ does not. This is done by moving one face of $\Omega_k(c)$ between c and $x^{(k)}$. The face moved is orthogonal to the axis along which the component of $x^{(k)} - c$, normalized with respect to Ω, is maximal. That face is moved so that it passes through the point $Ac + (1 - A)x^{(k)}$, where $A \in (0, 1)$. Clearly $A = 0$ puts the face through $x^{(k)}$, and $A = 1$ through c. Only the control point's box is shortened; all other filter points' boxes are not

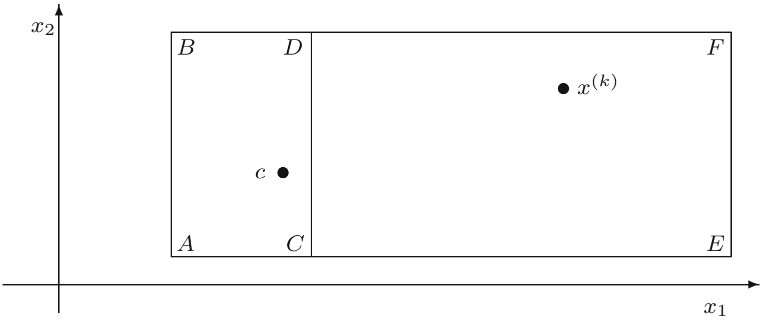

Fig. 3 The F- OSCARS cutting process: the box $\Omega_k(c)$ (shown as ABFE) is sampled at $x^{(k)}$. The point $x^{(k)}$ is rejected by the filter, and so $\Omega_k(c)$ is cut. The cut is orthogonal to the x_1 axis because $x^{(k)} - c$ has its longest normalized component along that axis. Cutting yields the new box $\Omega_{k+1}(c)$, which is ABDC

changed. This completes an update of the filter and associated list of boxes $\{\Omega_k(z) : z \in F\}$. An illustration of the cutting process is given in Fig. 3.

When $x^{(k)}$ is a better point than F_{k-1}, all boxes in the filter (including $\Omega_{k-1}(c)$) remain as they are. Since $x^{(k)}$ is a better point, there exists a $z \in F_{k-1}$ for which $x^{(k)} \prec z$. However it is not necessarily true that $x^{(k)} \prec c$. Hence $\Omega_{k-1}(c)$ is not shortened; to do so would exclude a known better point $x^{(k)}$ from $\Omega_k(c)$. The box for the new point $x^{(k)}$ is set equal to Ω.

If the box $\Omega_k(c)$ becomes too small, that box is reset to Ω via $\Omega_k(c) = \Omega$. The size $\|\Omega_k(c)\|_{\text{nd}}$ of $\Omega_k(c)$ is defined as the infinity norm of the normalized diagonal $u^{(k)}(c) - \ell^{(k)}(c)$. Specifically

$$\|\Omega_k(c)\|_{\text{nd}} = \max_{i=1,\ldots,n} \frac{u_i^{(k)}(c) - \ell_i^{(k)}(c)}{U_i - L_i} \tag{12}$$

which is the infinity norm of the scaled main diagonal of $\Omega_k(c)$, where the scaling maps Ω to the unit hypercube.

The next theorem shows that F- OSCARS satisfies Assumption 3d.

Theorem 9 *The sequence of iterates $\{x^{(k)}\}_{k=0}^{\infty}$ contains an infinite number of sample points drawn randomly from Ω with probability one.*

Proof Our approach is to show that drawing only a finite number of sample points from Ω (as opposed to those drawn from proper subsets of Ω) is a probability zero event. Let the K^{th} iteration be the last iteration at which $x^{(k)}$ is chosen randomly from Ω. We partition each filter F_k into two disjoint parts: H_k and $E_k = F_k - H_k$, where $z \in H_k$ if and only if $\Omega_k(z) = \Omega$. The probability that a point is drawn randomly from Ω at iteration k is $|H_k|/|F_k|$.

The only way a *new* point in $E_k - E_{k-1}$ can be found is generating a comparable or worse point in $\Omega_k(z) \equiv \Omega$ for some $z \in H_k$ and then shortening $\Omega_k(z)$. Once $k > K$ this is impossible. Hence $|E_k| \le |E_{k-1}|$ for all $k > K$, and the probability of drawing a point randomly from Ω is at least $p = 1/(1 + |E_K|)$ at any iteration for which H_k is non-empty and $k > K$. Thus it suffices to show that the number of non-empty H_k is infinite, almost surely.

Choose $T > K$ such that H_k is empty for all $k > T$. If a better point z is found at iteration k, this better point is included in both F_k and H_k. Pruning does not occur because z dominates another filter point, and $|F_k| \le |F_{k-1}|$. Hence no better point can ever be found for all $k > T$.

All other possibilities result in the control point's box being shortened. Since $|E_K|$ is finite, some $z \in E_K$ has its box shortened an infinite number of times. To avoid $\Omega_k(z)$ being reset to Ω in step 5, $\|\Omega_k(z)\|_{\text{nd}} > h_{\min}$ must hold for all $k > T$. Now $\|\Omega_k\|_{\text{nd}} \equiv \|\Omega_k\|_\infty$ in [15] when Ω is scaled to the unit hypercube. Theorem 2 of [15] shows that shortening a box an infinite number of times without triggering a reset to Ω is a probability zero event. □

Algorithm 2 *The filter version of* OSCARS.

1. Choose a random point $x^{(0)} \in \Omega$. Calculate $f_c = f(x^{(0)})$. Set $k = m = 0$, set $c = x^{(0)}$. Choose $\Theta \in (\tau, \Theta_{\max}]$ and $A \in (0, 1)$. Select $h_{\min} > 0$. Pick the maximum filter size $N \geq 3$. Set $F_0 = \{c\}$ and $\Omega_0(c) = \Omega$.
2. Increment k. If stopping conditions hold exit the algorithm. Randomly select first $c \in F_{k-1}$, then $x^{(k)} \in \Omega_{k-1}(c)$. Set $F_{\text{temp}} = F_{k-1} \cup \{x^{(k)}\}$ and $\Omega_k(x^{(k)}) = \Omega$. Remove all dominated points from F_{temp}, giving F_k.
3. Set $\Omega_k(x) = \Omega_{k-1}(x)$ for all $x \in F_{k-1} - \{c\}$. If $x^{(k)}$ is a better point than F_{k-1}, set $\Omega_k(c) = \Omega_{k-1}(c)$ and go to step 2.
4. Let i maximize $|x_i^{(k)} - c_i|/(U_i - L_i)$. If $c \in F_k$, update $\Omega_k(c)$ via:

   (a) If $x_i^{(k)} < c_i$, set $\Omega_k(c) = \{x \in \Omega_{k-1}(c) : x_i \geq (1 - A)x_i^{(k)} + Ac_i\}$; otherwise
   (b) set $\Omega_k(c) = \{x \in \Omega_{k-1}(c) : x_i \leq (1 - A)x_i^{(k)} + Ac_i\}$.

If $\|\Omega_k(c)\|_{\text{nd}} \leq h_{\min}$ reset $\Omega_k(c) = \Omega$ and increment m.

5. If $|F_k| > N$, then prune F_k and reduce Θ if necessary. Go to step 2.

4.1 Stopping conditions

The method halted when it had satisfied two conditions. The first of these is that it had located a point at which the measure of infeasibility θ did not exceed the maximum constraint violation tolerance τ. Once this first condition was satisfied the method set a mark f_{mark} equal to the best known objective function value over $\Phi(\tau)$. Each time it found a point in $\Phi(\tau)$ lower than $f_{\text{mark}} - f_{\text{acc}}$ it reset f_{mark} to this new lowest value. Here f_{acc} is the tolerance value on changes in f below which improvements in f are regarded as negligible. The method halted when K iterations were completed without finding a new f_{mark}, where $K = 2\zeta nN\lceil -\log_a(h_{\min})\rceil$. Here $a = 1/(1 - A)$ and the ceiling function $\lceil b \rceil$ returns the smallest integer not less than b. This value for K was chosen as follows. It takes at most $\lceil -\log_a(h_{\min})\rceil$ shortenings of *each* face in a pair of opposite faces of a box to reduce the distance between them to less than $h_{\min}$ when Ω is scaled to the unit hypercube. Noting that all filter points collectively have at most nN pairs of faces, this means each filter point will have on average at least ζ complete cycles through a range of box sizes from Ω to $h_{\min}$ assuming no unfiltered points are found. Due to the stochastic nature of the method, there is no guarantee that all points will actually pass through ζ cycles.

5 Numerical results

F- OSCARS was tested on 13 constrained global optimization test problems. The first 12 are from [11] and the 13th from [12]. They appear together in [19]. These results are compared with two other methods, both of which list results for these problems. The first of these [8] is the filter based simulated annealing method FSA which employs a variant of the Nelder Mead method as a local search. Runarsson and Yao [19] present an evolutionary method (SRES) which determines the fitness of individuals in a population by a stochastic bubble sort process.

The parameter values used by our method were $\tau = 10^{-6}$, $f_{\text{acc}} = 10^{-3}$, $\beta = 1.1$, $J_{\min} = -2$, $N = 30$, $A = 0.9$, $h_{\min} = 10^{-8}$, $\Theta_{\max} = \infty$, and $\zeta = 6$. These values give $K = 2880\,n$ for the stopping condition. We also used θ as defined in (3) to measure infeasibility. The squared term narrows the width of each band with $\theta > 1$ during pruning, which increases the emphasis on reducing θ when θ is large.

An examination of (9) shows the largest pruning mark is bounded below by $O(\beta^N \tau)$. If $\beta^N \tau$ is very large, the pruned filter is likely to contain highly infeasible points at all times. Intensively exploring around such points can be wasteful. Numerical experimentation verifies this, with $N = 60$, $\beta = 1.5$ and $N = 30$, $\beta = 2$ both being worse than the values given just above.

Table 1 compares results from our method (averaged over 30 runs), with those for [8] and [19]. The results for [8] and [19] have been lifted directly from those sources (Table 2 for [19]). The problems have been separated into two groups for analysis: those solved by all 3 methods, and those for which at least one method did not get an accurate estimate of the solution. For the former (g3, g4, g8, g9, g11, and g12) F- OSCARS used the fewest f evaluations on all six, and sometimes significantly so. For the remaining seven problems, F- OSCARS and SRES obtained the most accurate solution on three problems, and FSA on two (FSA and F- OSCARS tied on g6). FSA and F- OSCARS each solved two more problems to within 1 % of the best f value known, and SRES did so on one. It is worth noting that FSA [8] has the advantage of using a version of Nelder Mead as a local search procedure, in contrast to the other two methods. F- OSCARS found points with a constraint violation less than the maximum acceptable tolerance $\tau = 10^{-6}$ on all runs on all problems.

We also compare F- OSCARS with the fish swarm method P- BF AFS in [18]. The stopping rule used in [18] requires explicit knowledge of the optimal objective function value f^*: P- BF AFS halts when either a point achieving both feasibility and optimality within tolerance is found, or the maximum number of iterations is reached. Knowledge of f^* prevents early termination at a sub-optimal point, and avoids wasted function evaluations after optimality is achieved. F- OSCARS does not use f^* in its stopping rule (or elsewhere), which potentially puts F- OSCARS at a disadvantage. On the harder 7 problems, P- BF AFS did better than F- OSCARS on four, worse on two, and the same on one. They tied on g6, with F- OSCARS being faster on that problem. For g5 and g13 the solutions found by P- BF AFS exceeded our constraint violation tolerance of 10^{-6}. Our method was faster on five of the six easier problems; P- BF AFS was quicker on g3.

Table 5 of [1] lists final function values for an electromagnetic-like method, a particle swarm method, and a differential evolution method for problems g1-g13. Unfortunately function counts are not listed. On four of the seven harder problems, ours obtained a better objective function value than any method listed in [1]. Not all methods in [1] were able to solve all of the six easier problems.

We compared our method to the interval methods of Sun and Johnson [20] on their example 4, which is essentially problem 4.3 of [7]. Our algorithm was able to obtain an average best function value of −4.5142 over 40 runs using an average of 24,839 function evaluations per run. The better interval method of [20] obtained a best function value of −4.4828, and required more than 100,000 function evaluations and 18,000 inclusion function evaluations to do so. The global optimum is $f^* = -4.5142$.

DIRECT [9] solved the Gomez3 problem to within 1 % of the optimum after 89 function evaluations, and within 0.01 % of optimum after 513 function evaluations. In contrast F- OSCARS took an average of 282 function evaluations to reach a 1 % accuracy, and an average of 1369 function evaluations to reach 0.01 %, each averaged over 40 runs. DIRECT uses the auxiliary function $[f - f^*]_+ + \sum w_i \max(g_i, 0)$ where the w_i are positive weights and f^* is

Table 1 F- OSCARS, FSA [8] and SRES [19] are compared on problems g1-13 of Michalewicz et al. [11,12] (also listed in [19])

	n	q_i	q_e	Best known f	F- OSCARS		FILTER SIM. ANNEAL. (FSA)			SRES	
					Avg f	Avg nf	Avg f	Avg nf	Avg nc	Avg f	Avg nf
g1	13	9	0	−15	−13.6915	188,374	−14.9933	205,748	87,701	**−15.000**	148,200
g2	20	2	0	−0.803619	−0.638195	165,608	−0.371708	227,832	101,903	**−0.781975**	217,200
g5	4	2	3	5126.498	5136.90	175,548	**5126.498**	47,661	17,757	5128.881	51,600
g6	2	2	0	−6961.814	**−6961.81**	47,264	**−6961.81**	44,538	15,817	−6875.94	118,000
g7	10	8	0	24.306	24.4183	171,405	24.3795	404,501	171,299	**24.374**	143,000
g10	8	6	0	7049.331	**7369.19**	345,685	7509.32	243,520	93,667	7559.152	128,400
g13	5	0	3	0.053950	**0.0541050**	62,125	0.297720	120,268	42,268	0.067543	69,800
g3	10	0	1	−1.000	−0.999993	**64,222**	−0.999187	314,938	118,404	−1.000	229,200
g4	5	6	0	−30,665.539	−30,665.539	**39,678**	−30,665.5	86,154	37,000	−30,655.539	88,200
g8	2	2	0	−0.095825	−0.095825	**7740**	−0.095825	56,476	23,219	−0.095825	76,200
g9	7	4	0	680.630	680.639	**71,731**	680.636	324,569	147,035	680.656	111,400
g11	2	0	1	0.750000	0.749999	**8493**	0.749999	23,722	8485	0.750	11,400
g12	3	1	0	−1.00000	−1.000000	**8898**	−1.00000	59,355	25,818	−1.000000	16,400

Here n, q_i and q_e are the dimension, and numbers of inequality and equality constraints for each problem. Columns headed ‘avg f’, ‘avg nf’ and (for FSA) ‘avg nc’ list 30 run averages of the objective function value, and averages of the objective and constraint function counts. Not all methods solved the first seven listed problems; for these problems the most accurate solutions are marked in bold. All methods solved the last six problems; the method which evaluated f least often on average is marked in bold. On each problem F- OSCARS found a point which was feasible to within the constraint violation tolerance $\tau = 10^{-6}$ on every run the global minimum. DIRECT works by generating increasingly fine covers of Ω, where each cover consists of hyper-rectangles (or boxes) aligned with the coordinate axes. The values of f and g are known at the centre point of each box. At each iteration some boxes are subdivided into three equal sized boxes. DIRECT avoids having to know f^* by subdividing all boxes which would be selected for some possible value of f^*. On Gomez3 DIRECT is clearly quicker.

Regis and Shoemaker [16] also solve Gomez3 to one percent accuracy in f, their methods taking 53 and 30 function evaluations. Whilst this is clearly impressive, their methods model the objective function using radial basis functions. They require the solution of a maximin problem over the feasible region, and minimize an approximation of the objective over the feasible region at each iteration. Radial basis function methods are highly suited to problems with computationally expensive objective or constraint functions. In contrast methods such as F- OSCARS, with its lower computational overheads, are better suited to problems for which f and g can be calculated cheaply. Interestingly, sub-problems of radial basis methods such as [16] have this property, and F- OSCARS could be used to solve such sub-problems.

As it stands, F- OSCARS suffers from the curse of dimensionality because it only cuts one side of a box at a time. This is a quirk of F- OSCARS, not an inherent feature of the filter approach. This curse could easily be avoided by modifying F- OSCARS so that, for example, half the faces of a box were cut at each iteration, when n is large.

6 Conclusion

A filter based template for non-smooth black-box bound and otherwise constrained global optimization has been presented, and convergence shown almost surely under mild conditions. A variation of OSCARS has been implemented. This method is relatively simple, and has constant overheads per sample point. It is best suited to objective and constraint functions which are not too expensive to evaluate. Numerical results show this method is competitive in practice. There are many unexplored methods conforming to the filter template, both within the relatively new class of ARS algorithms, and elsewhere.

Acknowledgments The authors would like to thank two anonymous referees for their comments, which have improved the paper.

References

1. Ali, M.M., Golalkhani, C.M., Zhuang, J.: A computational study of different penalty approaches for solving constrained global optimization problems with the electromagnetism-like method. Optimization **63**, 403–419 (2014). doi:10.1080/02331934.2012.655691
2. Ali, M.M., Storey, C.: Topographical multi-level single linkage. J. Glob. Optim. **5**, 349–358 (1994)
3. Appel, M.J., Labarre, R., Radulović, D.: On accelerated random search. SIAM J. Optim. **14**, 708–731 (2003)
4. Birgin, E.G., Floudas, C.A., Martinez, J.M.: Global minimization using an augmented Lagrangian method with variable lower-level constraints. Math. Prog. Ser. A. **125**, 139–162 (2010). doi:10.1007/s10107-009-264-y
5. Di Pillo, G., Lucidi, S., Rinaldi, F.: An approach to constrained global optimization based on exact penalty functions. J. Glob. Optim. **54**, 251–260 (2012). doi:10.1007/s10898-010-9582-0
6. Fletcher, R., Leyffer, S.: Nonlinear programming without a penalty function. Math. Program. Ser. A **91**, 239–269 (2002)
7. Floudas, C.A., Pardalos, P.M.: A Collection of Test Problems for Constrained Global Optimization Problems, Lecture Notes in Computer Science vol. 455. Springer, Berlin (1990)

8. Hedar, A.-R., Fukushima, M.: Derivative free filter simulated annealing method for constrained continuous global optimization. J. Glob. Optim. **35**, 521–549 (2006). doi:10.1007/s10898-005-3693-z
9. Jones, D.: Direct global optimization algorithm. In: Floudas, C.A., Pardalos, P.M. (eds.) Encyclopedia of Optimization, vol. 1. Kluwer, Dordrecht, pp. 431–440 (2001)
10. Jones, D., Perttunen, C.D., Stuckman, B.E.: Lipschitzian optimization without the Lipschitz constant. J. Optim. Theory Appl. **79**, 157–181 (1993)
11. Koziel, S., Michalewicz, Z.: Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. Evol. Comput. **7**, 19–44 (1999)
12. Michalewicz, Z.: Genetic algorithms, numerical optimization and constraints. In: Eshelman, L.J. (ed.) Proceedings of 6th Internatinal Conference Genetic Algorithms. Morgan Kaufman, San Mateo, pp. 151–544 (1995)
13. Pedamallu, C.S., Özdamar, L., Czendes, T., Vinkó, T.: Efficient interval partitioning for constrained global optimization. J. Glob. Optim. **42**, 369–384 (2008). doi:10.1007/s10898-008-9297-7
14. Price, C.J., Reale, M., Robertson, B.L.: A cover partitioning method for bound constrained global optimization. Optim. Methods Softw. **27**, 1059–1072 (2012). doi:10.1080/10556788.2011.557726
15. Price, C.J., Reale, M., Robertson, B.L.: One side cut accelerated random search: a direct search method for bound constrained global optimization. Optim. Lett. **8**, 1137–1148 (2014). doi:10.1007/s11590-013-0631-8
16. Regis, R.D., Shoemaker, C.A.: Constrained global optimization of expensive black box functions using radial basis functions. J. Glob. Optim. **31**, 153–171 (2005)
17. Kan, Rinnooy: A.H.G.R., Timmer, G.T.: Stochastic global optimization methods part II: multi-level methods. Math. Program. **39**, 57–78 (1987)
18. Rocha, A.M.A.C., Costa, M.F.P., Fernandes, E.M.G.P.: A filter-based fish swarm algorithm for constrained global optimization: theoretical and practical issues. J. Glob. Optim. **60**, 239–263 (2014). doi:10.1007/s10898-014-0157-3
19. Runarsson, T.P., Yao, X.: Stochastic ranking for constrained evolutionary optimization. IEEE Trans. Evol. Comput. **4**, 284–294 (2000)
20. Sun, M., Johnson, A.W.: Interval branch and bound with local sampling for constrained global optimization. J. Glob. Optim. **33**, 61–82 (2005). doi:10.1007/s10898-004-6097-6
21. Vaz, A.I.F., Vicente, L.N.: A particle swarm pattern search method for bound constrained global optimization. J. Glob. Optim. **39**, 197–219 (2007). doi:10.1007/s10898-007-9133-5
22. Wang, C.-Y., Li, D.: Unified theory of augmented Lagrangian methods for constrained global optimization. J. Glob. Optim. **44**, 433–458 (2009). doi:10.1007/s10898-008-9347-1
23. Wu, Z.Y., Bai, F.S., Lee, H.W.J., Yang, Y.J.: A filled function method for constrained global optimization. J. Glob. Optim. **39**, 495–507 (2007). doi:10.1007/s10898-007-9152-2
24. Wu, Z.Y., Bai, F.S., Yang, Y.J., Mammadov, M.: A new auxiliary function method for general constrained global optimization. Optimization **62**, 193–210 (2013). doi:10.1080/02331934.2011.571258