Stopping Rules Effect on a Derivative-Free Filter Multistart Algorithm for Multilocal Programming

Florbela P. Fernandes[1,3], **M. Fernanda P. Costa**[2,3], **Edite M.G.P. Fernandes**[4]

[1] *Department of Mathematics, ESTiG- Institute Polytecnic of Bragança, Campus de Santa Apolónia, 5301-857 Bragança, Portugal* fflor@ipb.pt
[2] *Department of Mathematics and Applications, University of Minho, Guimarães, Portugal* mfc@math.uminho.pt
[3] *Centre of Mathematics, University of Minho, 4710-057 Braga, Portugal*
[4] *Algoritmi R&D Centre, University of Minho, 4710-057 Braga, Portugal* emgpf@dps.uminho.pt

Extended Abstract

1 Introduction

Multilocal programming aims to identify all the local (global and non-global) solutions of constrained nonlinear optimization problems and it has a wide range of application in engineering field. The purpose of this paper is to analyze the effect of stopping rules on the performance of a particular multistart method, which relies on a derivative-free local search procedure to converge to a solution, when solving multilocal optimization problems. The method herein presented implements a filter methodology to handle the constraints by forcing the local search towards the feasible region. The problem to be addressed is of the following type:

$$\begin{array}{rll} \min & f(x) & \\ \text{subject to} & g_j(x) \leq 0, & j = 1, ..., m \\ & l_i \leq x_i \leq u_i, & i = 1, ..., n \end{array} \tag{1}$$

where, at least one of the functions $f, g_j : \mathbb{R}^n \longrightarrow \mathbb{R}$ is nonlinear and $\text{F} = \{x \in \mathbb{R}^n : l_i \leq x_i \leq u_i, i = 1, \ldots, n, g_j(x) \leq 0, j = 1, \ldots, m\}$ is the feasible region. This kind of problems may have many global/local optimal solutions and so, it is important to develop a methodology that is able to explore the entire search space and find all the minima guaranteeing, in some way, that convergence to a previously found minimum is avoided.

2 The Proposed Filter Multistart Method

The methodology used to solve the problem (1) is a multistart algorithm coupled with a clustering technique to avoid the convergence to already detected solutions. Our proposal for the local search, a crucial procedure inside the multistart, relies on a technique that generates Approximate Descent Directions, without using derivatives, denoted by ADD method in [2]. In this study, the ADD method is combined with a (line search) filter method that aims at generating trial solutions that might be acceptable if they improve the constraint violation or the objective function.

2.1 The multistart strategy

Multistart is a stochastic algorithm where a local search is applied to a point, which is randomly generated in the search space, in order to converge to a local solution. To avoid detecting previously computed solutions, a clustering technique based on computing the regions of attraction of previously identified minima is applied. The region of attraction A_i of a local minimizer y_i associated with a local search procedure $\mathbf{L}$ is defined as $A_i = \{x \in \mathrm{F} : y_i = \mathbf{L}(x)\}$. A multistart algorithm aims at invoking the local search procedure N times, where N is the number of local solutions of (1). Since the region of attraction A_i of each minimizer y_i is not easy to compute, a simple stochastic procedure is used to estimate the probability, p, that a randomly generated point will not belong to a set, which is the union of a certain number of regions of attraction, i.e., $p = P[x \notin \cup_{i=1}^{k} A_i]$. The probability p is estimated [3] taking into account that the maximum attractive radius of the minimizer y_i is defined by:

$$R_i = \max_j \left\{ \left\| x_i^{(j)} - y_i \right\| \right\}, \tag{2}$$

where $x_i^{(j)}$ are the generated points which (may) led to the minimizer y_i. Given a randomly generated point x, let $z = \frac{\|x-y_i\|}{R_i}$. If $z \leq 1$ then x is likely to be inside the region of attraction of y_i. On the other hand, if the direction from x to y_i is ascent then x is likely to be outside the region of attraction of y_i. Based on the suggestion in [4], an estimate of the probability that $x \notin A_i$ may be given by:

$$p(x \notin A_i) = \begin{cases} 1, & \text{if } z > 1 \text{ or the direction from } x \text{ to } y_i \text{ is ascent} \\ \varrho\, \phi(z,l), & \text{otherwise} \end{cases} \tag{3}$$

where l is the number of times y_i has been recovered so far, $\varrho \in [0,1]$ and the function $\phi(z,l)$ satisfies the properties:

$$\lim_{z\to 0} \phi(z,l) \to 0, \ \lim_{z\to 1} \phi(z,l) \to 1, \ \lim_{l\to\infty} \phi(z,l) \to 0 \text{ and } 0 < \phi(z,l) < 1.$$

The function $\phi(z,l) = z \exp\left(-l^2(z-1)^2\right)$, for all $z \in (0,1)$, is used as proposed in the Ideal Multistart method [4].

2.2 The derivative-free filter procedure

The Approximate Descent Direction Filter (ADDF) method, a filter methodology combined with the ADD method presented in [2], is proposed for the local search. The filter methodology incorporates the concept of nondominance, present in the field of multiobjective optimization, to build a filter that is able to accept a trial point if it improves either the objective function or the constraint violation, relative to the current point. In this way, problem (1) is reformulated as a biobjective problem involving the original objective function f and the constraint violation function θ defined by

$$\theta(x) = \sum_{i=1}^{m} \left(\max_i \{0, g_i(x)\} \right)^2 + \sum_{i=1}^{n} (\max\{0, x_i - u_i\})^2 + (\max\{0, l_i - x_i\})^2 . \tag{4}$$

The ADDF is an iterative method that is applied to a randomly generated point x and provides a trial point y that is an approximate minimizer of problem (1). The point y is computed based on a direction d and a step size $\alpha \in (0, 1]$ in such a way that

$$y = x + \alpha\, d. \tag{5}$$

The procedure that decides which step size is accepted to generate an acceptable approximate minimizer is a filter method. After a search direction d has been computed, a step size α is determined by a backtracking line search technique. A decreasing sequence of α values is tried until a set of acceptance conditions are satisfied. The trial point y, in (5), is acceptable if sufficient progress in θ or in f is verified, relative to the current point x, as shown:

$$\theta(y) \leq (1 - \gamma_\theta)\, \theta(x) \ \text{ or } \ f(y) \leq f(x) - \gamma_f\, \theta(x) \tag{6}$$

where $\gamma_\theta, \gamma_f \in (0, 1)$. However, when x is (almost) feasible, i.e., in practice when $\theta(x) \leq \theta_{\min}$, the trial point y has to satisfy only the condition

$$f(y) \leq f(x) - \gamma_f\, \theta(x) \tag{7}$$

to be acceptable, where $0 < \theta_{\min} \ll 1$. To prevent cycling between points that improve either θ or f, at each iteration, the algorithm maintains the filter $\mathcal{F}$ which is a set of pairs (θ, f) that are prohibited for a successful trial point. During the backtracking line search procedure, the y is acceptable only if $(\theta(y), f(y)) \notin \mathcal{F}$. If the trial point does not satisfy the stopping conditions of the algorithm, the process is repeated with $x \leftarrow y$.

The filter is initialized with pairs (θ, f) that satisfy $\theta \geq \theta_{\max}$, where $\theta_{\max} > 0$ is the upper bound on θ. Furthermore, whenever y is accepted because condition (6) is satisfied, the filter is updated, and all entries that are dominated by the new entry are withdrawn from the filter. When it is not possible to find a point y with a step size $\alpha > \alpha_{\min}$ ($0 < \alpha_{\min} << 1$) that satisfy one of the conditions (6) or (7), a restoration phase is invoked. In this phase, the algorithm recovers the best point in the filter and a new trial point is determined according to the strategy based on equation (5). The algorithm implements the ADD method [2] to compute the direction d, required in (5). If the current point is feasible, the direction d will be descent for f; otherwise will be descent for θ.

2.3 Stopping rules

Some stopping rules have been proposed in the past in a multistart paradigm context (see [3] and the references therein included). Since the stopping rules aim to identify multiple optimal solutions, they should exhibit special properties. They should stop the algorithm when all minima have been identified with certainty, and they should not require a large number of local searches to decide that all minima have been found. Two different stopping rules have been tested.

First Stopping Rule If s denotes de number of recovered local minima after having performed t local search procedures, then the estimate of the fraction of the uncovered space is given by $P(s) = \frac{s(s+1)}{t(t-1)}$ and the stopping rule is

$$P(s) \leq \epsilon \tag{8}$$

with ϵ being a small positive number [4].

Second Stopping Rule This scheme is based on probabilistic estimates for the number of times each of the minima is being rediscovered by the local search [3]. Let $L_1, L_2, \ldots, L_s$ be the number of local searches that converged to the corresponding local minima $x_1^*, x_2^* \ldots, x_s^*$, respectively; where x_1^* is discovered with only one local application of the local procedure. Further, let n_2 be the number of subsequent applications of the local search until x_2^* is found for the first time; and similarly for $n_3, n_4, \ldots, n_s$. That is, x_2^* is found after $1 + n_2$ local searches, x_3^* after $1 + n_2 + n_3$, and so on.

Then the expected number L_J^s of local search applications that have converged to x_J^*, at the time when the sth minimum is discovered for the first time, is

$$L_J^s = L_J^{s-1} + (n_s - 1)\frac{L_J}{\sum_{i=1}^s L_i} = L_J^{s-1} + (n_s - 1)\frac{L_J}{\sum_{i=1}^s n_i}, \; J \leq s-1, L_s^s = 1.$$

If K represents the number of local search performed without discovering any new minimum, after s minima have been found, then the expected number of times the Jth minimum is found at that moment, $\mathcal{L}_J^s(K)$, is obtained by

$$\mathcal{L}_J^s(K) = \mathcal{L}_J^s(K-1) + \frac{L_J}{K + \sum_{i=1}^s n_i} \text{ with } \mathcal{L}_J^s(0) = L_J^s$$

and thus the quantity $E_2(s, K) = \frac{1}{s}\sum_{J=1}^s \left((\mathcal{L}_J^s(K) - L_J)/(\sum_{i=1}^s L_i)\right)^2$ tends asymptotically to zero. A stopping rule based on the variance of E_2, $\sigma^2(E_2)$, will allow the algorithm to continue without finding new minima until the condition

$$\sigma^2(E_2) < \tau\sigma^2_{\text{last}}(E_2) \tag{9}$$

holds, where $\sigma^2_{\text{last}}(E_2)$ is the variance of E_2 computed at the time when the last minimum was retrieved and $0 < \tau < 1$.

3 Ilustrative Examples and Remarks

To perform a preliminary analysis on the behavior of the algorithm when different stopping rules are applied, classical optimization problems, with $n = 2$, multimodal objective functions and box and inequality constraints are considered. The filter multistart method was coded in MatLab and the results were obtained in a PC with an Intel(R) Core(TM)2 Duo CPU P7370 2.00GHz processor and 3 GB of memory. Since derivatives are not provided to the algorithm, the factor ϱ is estimated and set to 0.05. In the ADD Filter method, $\gamma_\theta = \gamma_f = 10^{-5}$, $\alpha_{\min} = 10^{-6}$, $\theta_{\min} = 10^{-3} \max\{1, 1.25\theta(x_{in})\}$, $\theta_{\max} = \max\{1, 1.25\theta(x_{in})\}$, where x_{in} is the initial point in the local search. For the tested stopping rules, we set $\epsilon = 0.06$ and $\tau = 0.5$ as suggested in [3]. Five minimization problems **P1**–**P5** are defined using different feasible regions.

P1: $\min f(x) \equiv \frac{1}{2}\sum_{i=1}^{2}(x_i^4 - 16x_i^2 + 5x_i)$, defined in the box $[-5, 5]^2$;

P2: similar to **P1** with an additional constraint $(x_1 + 5)^2 + (x_2 - 5)^2 - 100 \leq 0$;

P3: similar to **P2** with an additional linear constraint $-x_1 - x_2 - 3 \leq 0$;

P4: $\min f(x) \equiv \sum_{i=1}^{2} \sin(x_i) + \sin\left(\frac{2x_i}{3}\right)$, defined in the box $[3, 13]^2$;

P5: similar to P4 with an additional linear constraint $-2x_1 - 3x_2 + 27 \leq 0$.

Figure 1 shows the contours of $f(x)$ and constraint boundary of **P3** (on the left) and **P5** (on the right). Tables 1 and 2 list the results obtained when the two stopping rules are tested. Columns 2–5 represent average values obtained during the 30 executions: average number of identified minimizers "# min", average number of function evaluations "f.eval.", average time (in seconds) "time(s)", and average value of the obtained best global solutions "global". The last column represents the standard deviation of the obtained global solutions.

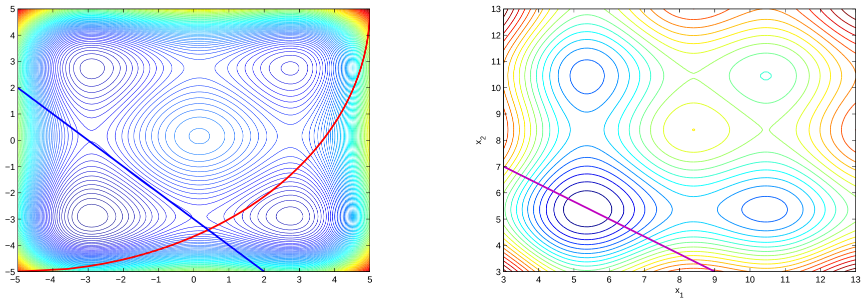

Figure 1: Objective function and constraints of **P3** and **P5**

When the first stopping rule is applied (8), the time is less than the time needed by the second stopping rule, in average. The time required by the algorithm increases with the number of function evaluations, which in turn depends on the number of local search calls. On the other hand, the stopping rule (9) is able to

Table 1: Numerical results obtained using the first stopping rule.

Prob.	# min	f.eval.	time(s)	global	S.D.
P1	3.9	4864.7	0.9	-78.3323	1.64525E-06
P2	3.8	9752.5	51.6	-78.3323	1.04324E-06
P3	3.1	13417.4	69.9	-64.1956	1.27448E-06
P4	3.9	4105.1	0.7	-2.4319	7.5900E-08
P5	4	7630.1	8.8	-2.4305	2.98961E-04

identify all the local solutions mostly. Further testing will be carried out specially with large-dimensional problems.

Table 2: Numerical results obtained using the second stopping rule.

Prob.	# min	f.eval.	time(s)	global	S.D.
P1	4	21326.4	6.8	-78.3323	1.82968E-06
P2	4	31670.6	235.3	-78.3323	1.13853E-06
P3	3.8	53014.1	149.6	-64.1956	7.01261E-07
P4	4	12974.2	2.2	-2.4319	5.06788E-08
P5	4	14336.9	11.5	-2.4305	1.04704E-04

Acknowledgments.

This work was financed by FEDER funds through COMPETE-Programa Operacional Fatores de Competitividade and by portuguese funds through FCT-Fundação para a Ciência e a Tecnologia within projects PEst-C/MAT/UI0013/2011 and FCOMP- 01-0124-FEDER-022674.

References

[1] M.F.P. Costa, E.M.G.P. Fernandes, Assessing the potential of interior point barrier filter line search methods: nonmonotone versus monotone approach, *Optimization*, **60**(10-11), pp. 1251–1268 (2011)

[2] A.R. Hedar, M. Fukushima, Derivative-Free Filter Simulated Annealing Method for Constrained Continuous Global Optimization, *Journal of Global Optimization*, **35**, pp. 521–549 (2006)

[3] I.E. Lagaris, I.G. Tsoulos, Stopping rules for box-constrained stochastic global optimization, *Applied Mathematics and Computation*, **197**, pp. 622–632 (2008).

[4] C. Voglis, I.E. Lagaris, Towards ”Ideal Multistart”. A stochastic approach for locating the minima of a continuous function inside a bounded domain, *Applied Mathematics and Computation*, **213**, pp. 1404–1415 (2009)