A penalty based filters method in direct search optimization

Aldina Correia
CIICESI / ESTG
P.PORTO
Felgueiras, Portugal
aic@estg.ipp.pt

João Matias
CM-UTAD
UTAD
Vila Real, Portugal
j_matias@utad.pt

Pedro Mestre
CITAB
UTAD
Vila Real, Portugal
pmestre@utad.pt

Carlos Serôdio
CITAB, Algoritmi
UTAD
Vila Real, Portugal
cserodio@utad.pt

***Abstract*—Constrained nonlinear optimization problems can be solved using penalty or barrier functions. This strategy, based on solving unconstrained problems obtained form the original problem, has shown to be effective, particularly when used with direct search methods. An alternative to solve the above mentioned problems is the filters method. The filters method, introduced by Fletcher and Leyffer in 2002, has been widely used to solve constrained problems. These methods use a different strategy when compared with penalty or barrier functions. The previous functions define a new one that combine the objective function and the constraints, while the filters method treat optimization problems as bi-objective problems where the objective function and a function that aggregates the constraints are optimized. Based on the work of Audet and Dennis, using filters method with derivative-free algorithms, the authors developed some works where other direct search methods were used, combining their potential with the filters method. More recently, a new variant of these methods was presented, where some alternative aggregation restrictions for the construction of filters were proposed. This paper presents a variant of the filters method, more robust than the previous ones, that has been implemented with a safeguard procedure where values of the function and constraints are linked together and are not treated as completely independently.**

***Index Terms*—Constrained nonlinear optimization, Filters method, Direct search methods.**

I. Introduction

A constrained NonLinear Problem (NLP) can be presented in the form:

$$\begin{array}{ll} \min & f(x) \\ s.t. & c_i(x) = 0, i \in \mathcal{E} \\ & c_i(x) \leq 0, i \in \mathcal{I} \end{array} \quad (1)$$

where, $x \in \mathbb{R}^n$, f is the objective function, $c_i(x) = 0$, $i \in \mathcal{E}$, with $\mathcal{E} = \{1, 2, ..., t\}$, define the equality constraints and $c_i(x) \leq 0$, $i \in \mathcal{I}$, with $\mathcal{I} = \{t+1, t+2, ..., m\}$ represent the inequality constraints.

We can define $\Omega = \{x \in \mathbb{R}^n : c_i = 0,\ i \in \mathcal{E} \wedge c_i(x) \leq 0, i \in \mathcal{I}\}$ as the set of all feasible points, i.e., the feasible region.

When the objective function and/or the constraints functions are not smooth, non continuous, it is not possible to use derivative-based methods. In these cases, we propose the use of derivative-free methods, more precisely, deterministic direct search methods, i.e., methods that only need and use information about the objective and constraints functions values to find the next iteration.

To deal with the constraints, using the most well known direct search methods (which are unconstrained optimization methods), we need some degree of constraints manipulation. The most frequent techniques are based in penalty or barrier functions. More recently, the filter methods has proved to be effective to deal with the information given by the constraints.

Unlike penalty and barrier methods, the filters method considers the feasibility and optimality separately, using the concept of dominance of multi-objective optimization. A filters algorithm introduces a function that aggregates constraint violations and constructs a bi-objective problem. It attempts to minimize simultaneously that function (*feasibility*) and the objective function (*optimality*), giving priority to the feasibility at least until a feasible iterate is found.

In short, we can say that in the resolution of a problem we have two objectives: minimize the objective function (Optimality) and minimize the constraints violation, which must be zero or tend to zero (Viability).

First filters method for derivative-free nonlinear programming was presented by Audet and Dennis, [4]. This method is based on pattern search methods. Motivated by this work the authors have developed a method that combines the features of the simplex method and filters method, [1]–[3]. The promising results that were obtained with this method encouraged the development of more features of the method, namely the combination of filters method with other direct search unconstrained optimization methods and the definition of other techniques to aggregate the constraint violation functions. This study was presented in [5].

In this paper, the fundamental concepts that allowed us to make this work, using the results obtained in [5], as a comparison with other implementations of the filters method, are presented.

II. Theoretical concepts

A key component of the filters method is a non-negative continuous function h which aggregates the constraint

ESTG – School of Management and Technology, P.PORTO – Polytechnic of Porto, CIICESI — Center for Research and Innovation in Business Sciences and Information Systems, UTAD – University of Trás-os-Montes and Alto Douro, CM-UTAD – Centre for Mathematics, CITAB – Centre for the Research and Technology of Agro-Environmental and Biological Sciences violation functions. Then h is a function such that $h(x) \geq 0$ with $h(x) = 0$ if and only if x is feasible.

This function is used in the definition of successive filters along the iterative process, because a point is accepted in a filter if and only if that point has better values of h or f than the points found so far.

Another fundamental concept to the perception of the filter method, is the dominance.

A point $x \in \mathbb{R}^n$ is said to *dominate* $y \in \mathbb{R}^n$, written $x \prec y$, if $f(x) \leq f(y)$ and $h(x) \leq h(y)$ or $f(x) < f(y)$ or $h(x) < h(y)$.

A *filter*, denoted by $\mathcal{F}$, is a finite set of points in the domain of f and h such that no point x in the set dominates other point y in the set, i.e., there is no pair of points x and y in the filter that have the relation $x \prec y$.

Figure 1, based on Correia et. al. [5] and Ribeiro et. al. [6], depicts the concept of a filter with four initial points (a, b, c and d).

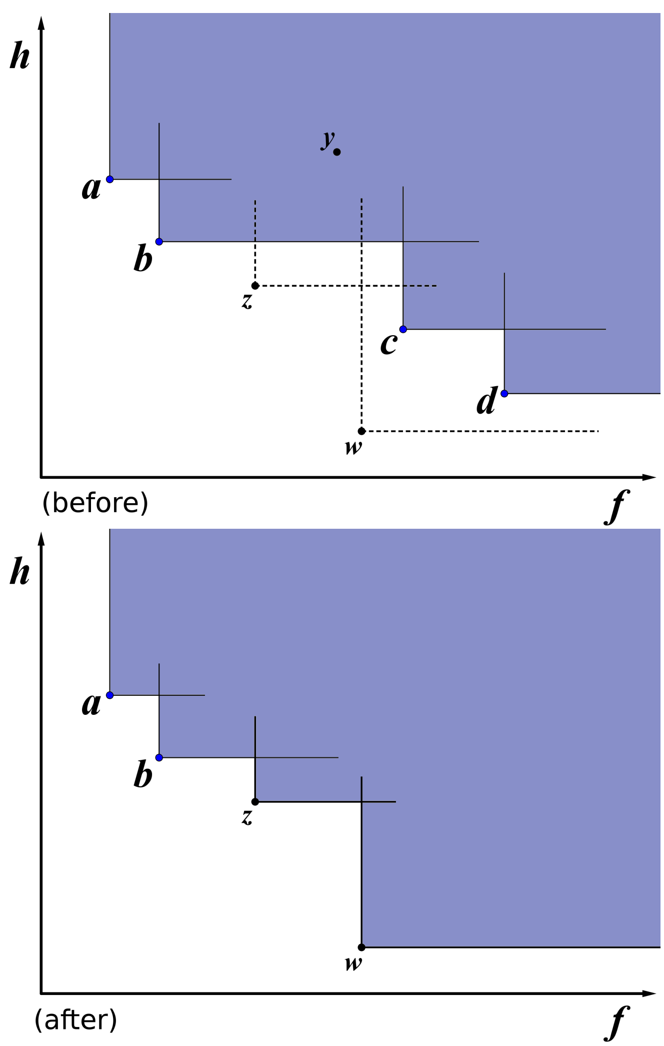

Fig. 1. Filters Method - Graphic Concept

Points represented by a, b, c and d define a boundary of the *forbidden region*, presented in shaded. To the filter it should be added the points with better (lower) values of f and h, i.e. the aim is to have $h = 0$ and the lowest possible values for f. Therefore the point represented by y, as it is inside the forbidden region, will not be accepted in the filter. But the point represented by z is out of the forbidden region and therefore it will be included in the filter. The same applies to the point represented by w , however, in this case, there would still rise to the elimination of points represented by c and d of the filter, since they are in the forbidden region defined by w , i.e., c and d are dominated by w .

Now follows another concept that is also important.

It is considered that a point x is *filtered* by a filter $\mathcal{F}$ if:

- There exists a point $y \in \mathcal{F}$ such that $y \prec x$ or $y = x$;
- or $h(x) \geq h_{max}$;
- or $h(x) = 0$ and $f(x) \geq f^F$;

where f^F is the objective function value of the best feasible point found so far and h_{max} is a previous defined bound for h value, so each point $x \in \mathcal{F}$ satisfies $h(x) < h_{max}$.

III. Filters method algorithm

Based on the algorithms presented in [4], [6]–[8], defined in a general manner, the authors have implemented and tested several versions of the filters method ([2] and [1]).

In these versions, the filters method was implemented in combination with the Hooke and Jeeves method, a pattern search method as Audet and Dennis and with Nelder-Mead method. In [3] some improvements were presented and a comparison was made of a new simplex filter algorithm with the first version of the same method.

Numerical results obtained have motivated the generic implementation of filters method, i.e. so that it can be applied not only with Nelder-Mead and Hooke and Jeeves methods, in optimization of h and f , as well as all available direct search type methods. That was presented in [5].

The present work uses the procedure implemented in [5] with changes, adaptations and generalizations of those methods.

While in previous work the filters method treats the optimality, optimization of f , completely isolated from admissibility, optimization of h , in this work there is a link between them, attempting that both processes not to be fully independent.

After some analysis, it was found that in the previous implementation there could occur cases of alternating values without obtaining convergence of the algorithm. This could occur on an iteration of a process to obtain a value, from which by the other process will obtain the the initial value.

The new implementation is depicted in Fig. 12 and the explanations are noted below.

In order to compare in a correct way, the same test problems, the same aggregate functions h , and the same direct testing methods as described in [5] were used.

Algorithm

The procedure begins with an initial filter that contains the initial iteration, $F_0 = x_0$. Then, it is constructed an initial Set (S_k) containing $n + 1$ points from that iteration (x_k) and: $S_k = \{x_k\} \cup \{x_k + e_i, i = 1, ..., n\}$, where $e_i, i = 1, ..., n$ represents the vectors of the canonic basis in $\mathbb{R}^n$, starting with the Search Set $i = 0, ..., n$.

1) If the point under analysis is feasible then its inclusion in the filter is evaluated:
   a) If it is not accepted:
      i) One of five unconstrained optimization methods is applied to the function F;
      ii) A new point is obtained, x_k;
      iii) Go back to the construction of the Set: $S_k = \{x_k\} \cup \{x_k + e_i, i = 1, ..., n\}$;
   b) If it is accepted:
      i) Filter is updated with the new approximation to the solution, i.e., the new iteration;
      ii) If the stop criterion is verified, this approximation is the solution. Otherwise, go back to the Set construction, using this point.
2) If the point is an infeasible one, its inclusion in the filter is evaluated:
   a) If it is not accepted:
      i) One of five unconstrained optimization methods is applied to the function h;
      ii) A new point is obtained, x_k;
      iii) Go back to the to the construction of the Set/Simplex : $S_k = \{x_k\} \cup \{x_k + e_i, i = 1, ..., n\}$;
   b) If it is accepted:
      i) The filter is updated with the new approximation to the solution, i.e., the new iteration;
      ii) If the stop criterion is verified, this approximation is the solution. Otherwise, go back to the Set construction, using this point.

Thus, the method contains two distinct processes: the *external iterative process*, involving the Set construction and the filter update and the *internal iterative process*, involving the optimization of F and h, where unconstrained optimization problems are solved, with objective functions f or h, using one of the Direct Search methods. These are de same methods described in [5], with which performance comparisons were made.

The main difference between this work and the one presented in [5] is the use of the F function, instead of the exclusively objective function f, from problem (1) as shown in 1.(a)i. from the above procedure.

The idea behind this new implementation is to construct the function F using not only the objective function f from the initial problem (1), but also the function h that may be used to aggregate constraint violations. This is illustrated in the formulation of the problem (2), where $F(x)$ is defined by $F(x) = f(x) + \eta h(x)$, resulting in the problem without constraints

$$\min_{x \in \mathbb{R}^n} \; F(x), \tag{2}$$

where η is a positive factor.

As in [5], the same five methods were used in internal process: Opportunistic Coordinate search method (CS); Hooke and Jeeves method (HJ); A version of Audet et. al. method (AA); Nelder-Mead method (NM) and a Convergent Simplex method (SC). The first three are Pattern Search Methods or Directional Direct-Search Methods.

The last two are Simplex Methods or Simplicial Direct-Search Methods.

IV. Aggregate constraint violation functions

Considering the problem (1) constraints, namely the t equality constraints, which may be written as two inequality constraints:

$$\begin{aligned} c_i(x) = 0,\ i = 1, ..., t \quad &\Leftrightarrow \quad c_i(x) \le 0 \wedge c_i(x) \ge 0,\ i = 1, ..., t \\ &\Leftrightarrow \quad c_i(x) \le 0 \wedge -c_i(x) \le 0,\ i = 1, ..., t \end{aligned}$$

settle $2t + m = n$, this can be rewritten defining:

$$\begin{cases} r_i(x) = c_i(x) \le 0, \quad i = 1, ..., t \\ r_j(x) = -c_i(x) \le 0, \quad j = t+1, ..., 2t; \quad i = 1, ..., t \\ r_j(x) = c_i(x) \le 0, \quad j = 2t+1, ..., q; \quad i = t+1, ..., m \end{cases}$$

the problem to solve will be:

$$\begin{array}{cl} \min\limits_{x \in \mathbb{R}^n} & f(x) \\ s.t. & r_i(x) \le 0, i = 1, ..., q \end{array} . \tag{3}$$

To construct the h function (function that aggregate the constraint violation) the norm 2 is usually used,

$$h(x) = \|C_+(x)\|_2 = \sqrt{\sum_{i=1}^{q} \max(0, r_i(x))^2}.$$

The requirements for h are: be continuous and $h(x) \ge 0$ with $h(x) = 0$ if and only if x is feasible, i.e., h must be a non negative continuous function which $h(x) = 0$ if and only if x is feasible. Therefore we propose the same alternatives presented in [5], to aggregate the constraint violation functions, and in that way, we can compare properly the obtained values.

The definitions of h used to aggregate the constraint violation functions, are presented in Table I.

V. Used parameters

In both processes, *internal* (Unconstrained Optimization - Direct Search Methods) and *external* (Constrained Optimization - Filters Method), it is necessary to choose some parameters. Once again, to compare the values obtained with those obtained in [5] same parameters were uses, where possible. The used parameters are presented in Tables III and II.

VI. Numerical Results

The test problems are the same as used in [5] and were selected from Schittkowski [9] and CUTE [10] collections. The fifteen Schittkowski problems are: S224; S225; S226; S227; S228; S231; S233; S234; S249; S264; S270; S323; S324; S325 and S326 and of Cute collection were chosen two test problems: C801 and C802. The last problem is the PA problem presented in [2].

The choice of these eighteen test problems was not made in accordance with any special requirement, they are only used to illustrate the performance of the methods implemented.

In order to classify the solution approximations, we use the same criteria used in [5], without using the **Bad** classification:

- a *Feasible solution approximation* if $h(x_k) = 0$, is:
  - *Good* if: $|f(x^*) - f(x_k)| \leq 0.0001$;
  - *Medium* if: $0.0001 < |f(x^*) - f(x_k)| \leq 0.01$;
- an *Infeasible solution approximation* if $h(x_k) \neq 0$, is:
  - *Good* if: $|f(x^*) - f(x_k)| \leq 0.0001 \wedge h(x_k) \leq 0.0001$;
  - *Medium* if:
    - $0.0001 < |f(x^*) - f(x_k)| \leq 0.01 \wedge h(x_k) \leq 0.0001$;
    - or $|f(x^*) - f(x_k)| \leq 0.0001 \wedge 0.0001 < h(x_k) \leq 0.01$;
    - or $0.0001 < |f(x^*) - f(x_k)| \leq 0.01 \wedge 0.0001 < h(x_k) \leq 0.01$;

All the obtained solution approximations were classified using these criteria.

In the above tests, it can be shown an improvement in the results, when compared with those presented in [5]. A table which summarizes the obtained results are presented in the left table of Fig. 2 and the differences to the results obtained in [5] are presented in the table on the right.

An improvement on the results is noticed in almost all types of solutions obtained above, i.e. in the good and medium ones, and for almost all direct search methods and penalty functions.

In order to enable an easier data analysis, it is present an analysis for each method and for each penalty function, based on right table of Fig. 2.

Methods		Solution Approximation							
		Feasible				Infeasible			
IP	EP	Good	Med	Total	%	Good	Med	Total	%
CS	N1	6		6	33,3%			0	0,0%
	N2	6		6	33,3%			0	0,0%
	NEB	7		7	38,9%			0	0,0%
	NP	3		3	16,7%			0	0,0%
HJ	N1	3		3	16,7%	2		2	11,1%
	N2	3	1	4	22,2%		1	1	5,6%
	NEB	4	2	6	33,3%			0	0,0%
	NP	1	2	3	16,7%		2	2	11,1%
AA	N1	5	1	6	33,3%		1	1	5,6%
	N2	4		4	22,2%		1	1	5,6%
	NEB	7	2	9	50,0%			0	0,0%
	NP	3	1	4	22,2%	2		2	11,1%
NM	N1	7		7	38,9%		1	1	5,6%
	N2	5		5	27,8%	1		1	5,6%
	NEB	10		10	55,6%			0	0,0%
	NP	3	2	5	27,8%	1	1	2	11,1%
SC	N1	5	2	7	38,9%	1	1	2	11,1%
	N2	7	1	8	44,4%	2		2	11,1%
	NEB	7		7	38,9%			0	0,0%
	NP	4	2	6	33,3%	1	2	3	16,7%
		100	16	116		10	10	20	

Methods		Solution Approximation							
		Feasible				Infeasible			
IP	EP	Good	Med	Total	%	Good	Med	Total	%
CS	N1	4	0	4	22,2%	0	0	0	0,0%
	N2	4	0	4	22,2%	-1	0	-1	-5,6%
	NEB	6	0	6	33,3%	0	0	0	0,0%
	NP	1	0	1	5,6%	-1	0	-1	-5,6%
HJ	N1	-2	0	-2	-11,1%	1	0	1	5,6%
	N2	-2	1	-1	-5,6%	-1	1	0	0,0%
	NEB	2	2	4	22,2%	0	0	0	0,0%
	NP	-3	2	-1	-5,6%	-1	2	1	5,6%
AA	N1	4	1	5	27,8%	0	1	1	5,6%
	N2	3	0	3	16,7%	0	1	1	5,6%
	NEB	5	2	7	38,9%	0	0	0	0,0%
	NP	2	1	3	16,7%	0	0	0	0,0%
NM	N1	5	0	5	27,8%	0	1	1	5,6%
	N2	3	0	3	16,7%	0	0	0	0,0%
	NEB	8	0	8	44,4%	0	0	0	0,0%
	NP	1	2	3	16,7%	0	1	1	5,6%
SC	N1	3	2	5	27,8%	0	1	1	5,6%
	N2	5	1	6	33,3%	0	0	0	0,0%
	NEB	5	0	5	27,8%	0	0	0	0,0%
	NP	2	2	4	22,2%	0	2	2	11,1%
		56	16	72		-3	10	7	

Fig. 2. Numerical Results

Thus, starting with the direct search methods, we have the results of the Coordinate Search method in Fig. 3. We can observe the large increase in the number of feasible solutions and a small decrease in infeasible solutions.

Methods		Solution Approximation							
		Feasible				Infeasible			
IP	EP	Good	Med	Total	%	Good	Med	Total	%
CS	N1	4	0	4	22,2%	0	0	0	0,0%
	N2	4	0	4	22,2%	-1	0	-1	-5,6%
	NEB	6	0	6	33,3%	0	0	0	0,0%
	NP	1	0	1	5,6%	-1	0	-1	-5,6%
	Total	**15**	**0**	**15**		**-2**	**0**	**-2**	
	%	16,7%	0,0%	16,7%		-2,2%	0,0%	-2,2%	

Fig. 3. Numerical Results-Coordinate Search

For the Hook-Jeeves method we have the results presented in Fig. 4. These were the worse results, where it was obtained fewer solutions than with the previous implementation in [5].

Methods		Solution Approximation							
		Feasible				Infeasible			
IP	EP	Good	Med	Total	%	Good	Med	Total	%
HJ	N1	-2	0	-2	-11,1%	-1	0	-1	-5,6%
	N2	-2	1	-1	-5,6%	-1	0	-1	-5,6%
	NEB	2	2	4	22,2%	0	0	0	0,0%
	NP	-3	2	-1	-5,6%	-1	0	-1	-5,6%
	Total	**-5**	**5**	**0**		**-3**	**0**	**-3**	
	%	-5,6%	5,6%	0,0%		-3,3%	0,0%	-3,3%	

Fig. 4. Numerical Results- Hooke-Jeeves

Results obtained with Audet et. al. method are presented in Fig. 5. Here were obtained better results than with the previous implementation for the feasible approximation. For the infeasible approximations it were obtained the same results with every penalty function.

Methods		Solution Approximation							
		Feasible				Infeasible			
IP	EP	Good	Med	Total	%	Good	Med	Total	%
AA	N1	4	1	5	27,8%	0	0	0	0,0%
	N2	3	0	3	16,7%	0	0	0	0,0%
	NEB	5	2	7	38,9%	0	0	0	0,0%
	NP	2	1	3	16,7%	0	0	0	0,0%
	Total	**14**	**4**	**18**		**0**	**0**	**0**	
	%	15,6%	4,4%	20,0%		0,0%	0,0%	0,0%	

Fig. 5. Numerical Results- Audet et. al.

For the Nelder-Mead method the results presented in Fig. 6 were obtained. The conclusions are similar to the previous case.

The results presented in Fig. 7 are for the Simplex Convergent algorithm. Once again, the conclusions are similar to the previous two cases.

For the aggregate constraints violation functions, there was also an performance increase in the problems resolution.

Thus, starting with the **N1** aggregate function , we have the results in Fig. 8. We can observe the large increase in the number of feasible solutions and also a small increase in infeasible solutions.

Methods		Solution Approximation							
		Feasible				Infeasible			
IP	EP	Good	Med	Total	%	Good	Med	Total	%
NM	N1	5	0	5	27,8%	0	0	0	0,0%
	N2	3	0	3	16,7%	0	0	0	0,0%
	NEB	8	0	8	44,4%	0	0	0	0,0%
	NP	1	2	3	16,7%	0	0	0	0,0%
	Total	**17**	**2**	**19**		**0**	**0**	**0**	
	%	18,9%	2,2%	21,1%		0,0%	0,0%	0,0%	

Fig. 6. Numerical Results- Nelder-Mead

Methods		Solution Approximation							
		Feasible				Infeasible			
IP	EP	Good	Med	Total	%	Good	Med	Total	%
SC	N1	3	2	5	27,8%	0	0	0	0,0%
	N2	5	1	6	33,3%	0	0	0	0,0%
	NEB	5	0	5	27,8%	0	0	0	0,0%
	NP	2	2	4	22,2%	0	0	0	0,0%
		15	**5**	**20**		**0**	**0**	**0**	
		16,7%	5,6%	22,2%		0,0%	0,0%	0,0%	

Fig. 7. Numerical Results- Simplex Convergent

For the **N2** aggregate function, we have the results in Fig. 9. The results are very similar to the previous case. Also, in both cases the Hooke-Jeeves method was the only one where it was not obtained better results.

The results for the **NEB** aggregate function, are presented in Fig. 10. With this aggregate constraints violation function, we obtained the better improvement of all results, particularly in the feasible solutions.

The results for the **NP** aggregate function, are presented in Fig. 11. With this aggregate constraints violation function, we obtained the lowest improvement of all results. Once again the poor performance of the Hooke-Jeeves resulted in a loss of solutions.

Methods		Solution Approximation							
		Feasible				Infeasible			
IP	EP	Good	Med	Total	%	Good	Med	Total	%
CS	N1	4	0	4	22,2%	0	0	0	0,0%
HJ		-2	0	-2	-11,1%	1	0	1	5,6%
AA		4	0	4	22,2%	0	0	0	0,0%
NM		5	0	5	27,8%	0	0	0	0,0%
SC		3	0	3	16,7%	1	0	1	5,6%
	Total	**14**	**0**	**14**		**2**	**0**	**2**	
	%	15,6%	0,0%	15,6%		2,2%	0,0%	2,2%	

Fig. 8. Numerical Results-N1 aggregate function

Methods		Solution Approximation							
		Feasible				Infeasible			
IP	EP	Good	Med	Total	%	Good	Med	Total	%
CS	N2	4	1	5	27,8%	-1	0	-1	-5,6%
HJ		-2	0	-2	-11,1%	-1	1	0	0,0%
AA		3	0	3	16,7%	0	0	0	0,0%
NM		3	0	3	16,7%	1	0	1	5,6%
SC		5	2	7	38,9%	2	1	3	16,7%
	Total	**13**	**3**	**16**		**1**	**2**	**3**	
	%	14,4%	3,3%	17,8%		1,1%	2,2%	3,3%	

Fig. 9. Numerical Results-N2 aggregate function

Methods		Solution Approximation							
		Feasible				Infeasible			
IP	EP	Good	Med	Total	%	Good	Med	Total	%
CS	NEB	6	0	6	33,3%	0	0	0	0,0%
HJ		2	2	4	22,2%	0	0	0	0,0%
AA		5	2	7	38,9%	0	0	0	0,0%
NM		8	0	8	44,4%	0	0	0	0,0%
SC		5	0	5	27,8%	0	0	0	0,0%
	Total	**26**	**4**	**30**		**0**	**0**	**0**	
	%	28,9%	4,4%	33,3%		0,0%	0,0%	0,0%	

Fig. 10. Numerical Results-NEB aggregate function

Methods		Solution Approximation							
		Feasible				Infeasible			
IP	EP	Good	Med	Total	%	Good	Med	Total	%
CS	NP	1	0	1	5,6%	-1	0	-1	-5,6%
HJ		-3	2	-1	-5,6%	-1	2	1	5,6%
AA		2	1	3	16,7%	2	0	2	11,1%
NM		1	2	3	16,7%	1	1	2	11,1%
SC		2	2	4	22,2%	1	2	3	16,7%
	Total	**3**	**7**	**10**		**2**	**5**	**7**	
	%	3,3%	7,8%	11,1%		2,2%	5,6%	7,8%	

Fig. 11. Numerical Results-NP aggregate function

For these 18 test problems, from various tested methods and constraints evaluation combination that that the best performance improvement was the NM combined with NEB. Besides being the combination with the best results, it was also the one that had the better improvements, when compared with our previous works.

VII. Conclusion

From the above presented numerical results it can be concluded that it is possible to use and improve other direct search methods and combining them with the filters method. Also, it is possible to improve the proposed technique for constraint violation functions aggregation.

In our particular case, it is predictable a significant improvement of the previously obtained results with the creation of a new objective function, by including a penalty term. This has proved to be an essential fact to the improvement of the results.

Thus, the suggestions presented in [5] together with the improvements proposed in this work, results in another alternative for solving constrained optimization problems without using derivatives of the functions involved or their approximations.

Acknowledgment

The authors would like to thank the financial support for participation in the conference to CIICESI – Center for Research and Innovation in Business Sciences and Information Systems (ESTG/P.Porto).

References

[1] A. Correia, J. Matias, P. Mestre, and C. Serdio, "Derivative-free optimization and filter methods to solve nonlinear constrained problems," *International Journal of Computer Mathematics*, vol. 86, no. 10, pp. 1841–1851, (2009).

[2]

[3] A. Correia, J. Matias, P. Mestre, and C. Serdio, "Derivative-free nonlinear optimization filter simplex," *International Journal of Applied Mathematics and Computer Science (AMCS)*, vol. 20, no. 4, (In Press, 2010).

[4] C. Audet and J. Dennis, "A pattern search filter method for nonlinear programming without derivatives," *SIAM Journal on Optimization*, vol. 5, no. 14, pp. 980–1010, (2004).

[5] A. Correia, J. Matias, P. Mestre, and C. Serdio, "Filters method in direct search optimization, new measures to admissibility," in *Proceedings of the 11th International Conference on Computational and Mathematical Methods in Science and Engineering*, vol. 1. CMMSE-Computational and Mathematical Methods in Science and Engineering, 2011, pp. 384–396.

[6] A. Ribeiro, E. Karas, and C. Gonzaga, "Global convergence of filter methods for nonlinear programming," *SIAM J. on Optimization*, vol. 19, no. 3, pp. 1231–1249, 2008.

[7] R. Fletcher, "Nonlinear programming without a penalty function," *Mathematical Programming*, vol. 91, pp. 239–269, 1997.

[8] E. W. Karas *et al.*, "Exemplos de trajetória central mal comportada em otimização convexa e um algoritmo de filtros para programação não linear," 2002.

[9] K. Schittkowski, *More Test Examples for Nonlinear Programming Codes, Economics and Mathematical Systems*. Berlin: Springer-Verlag, (1987).

[10] I. Bongartz, A. Conn, N. Gould, and P. Toint, "Cute: Constrained and unconstrained testing environment," *ACM Transactions and Mathematical Software*, no. 21, pp. 123–160, (1995).

APPENDIX

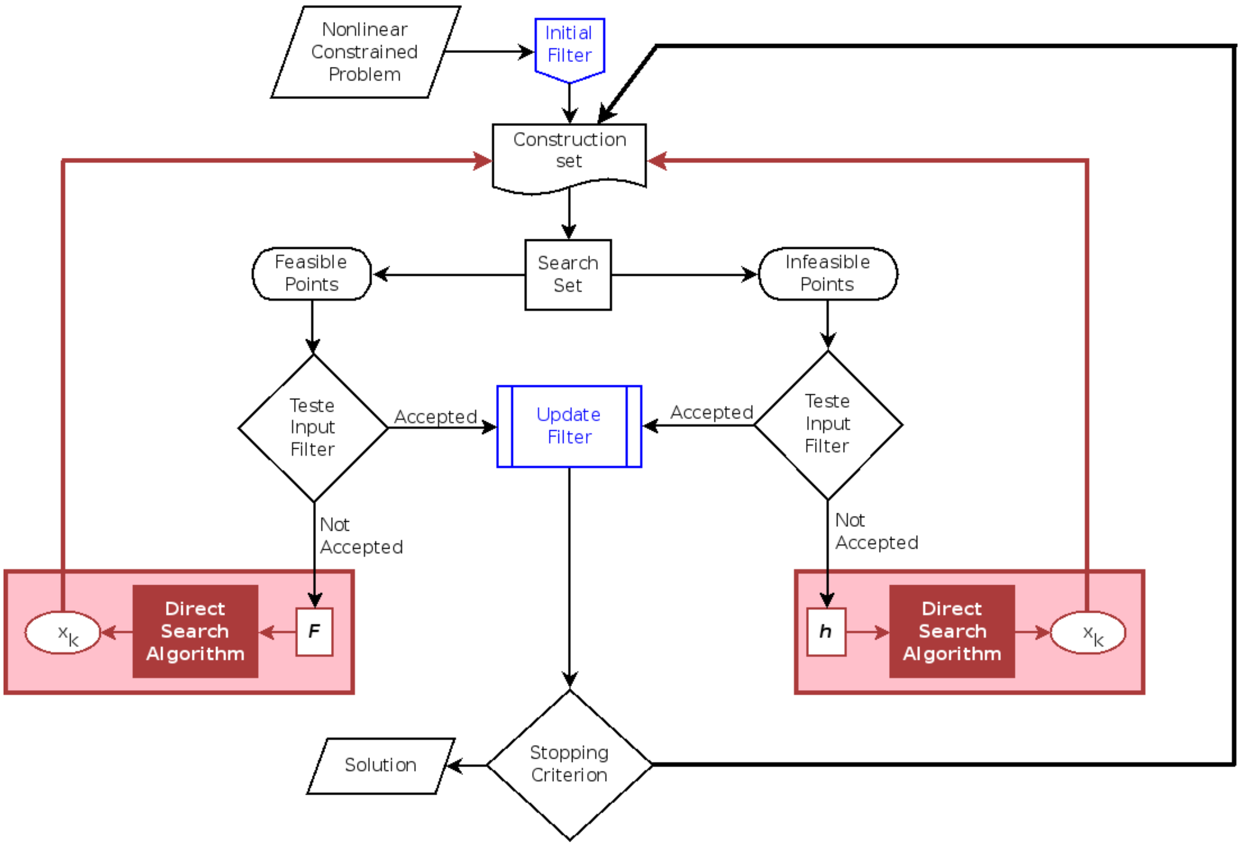

Fig. 12. Filters Method Algorithm

TABLE I
ALTERNATIVES TO AGGREGATE THE CONSTRAINT VIOLATION FUNCTIONS

Measure		h
Norm $1/\ell_1$ Penalty	**N1**	$h(x) = \lVert C_+(x)\rVert_1 = \sum_{i=1}^{q} \max[0,\ r_i(x)]$
Norm 2	**N2**	$h(x) = \lVert C_+(x)\rVert_2 = \sqrt{\sum_{i=1}^{q} \left\{\max[0,\ r_i(x)]^2\right\}}$
Extreme Barrier	**NEB**	$h(x) = \begin{cases} 0 & \text{if} \quad x \in \Omega \\ +\infty & \text{if} \quad x \notin \Omega \end{cases}$
Progressive Barrier Classic Penalty Static/Dynamic Penalty	**NP**	$h(x) = \sum_{i=1}^{q} \{\max[r_i(x),\ 0]\}^2$

TABLE II
CONSTRAINED OPTIMIZATION - FILTERS METHOD - PARAMETERS USED

$k_{max} = 40 \rightarrow$ Maximum number of iterations in the external process;
$\rho = 1 \rightarrow$ Initial search step length;
$T1 = \lvert x_k - x_{k+1} \rvert = 0.00001 \rightarrow$ tolerance for the distance between two consecutive iterations;
$T2 = \lvert f(x_k) - f(x_{k+1}) \rvert = 0.00001 \rightarrow$
Tolerance between 2 values of the objective function in two consecutive iterations;
$h_{max} = +\infty \rightarrow$ Maximal valor of constraints violation.

TABLE III
UNCONSTRAINED OPTIMIZATION - DIRECT SEARCH METHODS - USED PARAMETERS

Parameters	**Coordinate Search**	**Hooke-Jeeves**	**Audet**	**Nelder-Mead**	**Simplex Convergent**
k_{max}	100	100	100	100	100
s	1	1	*	1	1
s_m	*	*	1,5	*	*
s_p	*	*	1	*	*
s_{min}	10^{-3}	10^{-3}	10^{-3}	*	*
α	*	*	*	1	1
β	*	*	*	0,5	0,5
γ	*	*	*	2	2
T_1	10^{-5}	10^{-5}	10^{-5}	10^{-5}	10^{-5}
T_2	10^{-5}	10^{-5}	10^{-5}	10^{-5}	10^{-5}
T_{var}	*	*	*	10^{-5}	10^{-5}
T_{vol_n}	*	*	*	*	10^{-5}

$k_{max} \rightarrow$ Maximum number of iterations; $s \rightarrow$ Length of the initial step
$s_m \rightarrow$ Length of the initial mesh search step (Audet); $s_p \rightarrow$ Length of the initial poll step (Audet)
$s \rightarrow$ Length of the initial step; $s_{min} \rightarrow$ Minimum value for the step length
$\alpha \rightarrow$ Reflexion parameter (Nelder-Mead); $\beta \rightarrow$ Contraction parameter (Nelder-Mead)
$\gamma \rightarrow$ Expansion parameter (Nelder-Mead)
$T1 = \lvert x_k - x_{k+1} \rvert \rightarrow$ Tolerance for the distance between two consecutive iterations
or Tolerance for the distance between the last iteration and the latest iteration (Nelder-Mead)
$T2 = \lvert f(x_k) - f\left(x_{k+1}\right) \rvert \rightarrow$ Tolerance for
the distance between two values of the objective function in successive iterations
$T_{var} \rightarrow$ Tolerance to the variance of the objective function values in the vertices of the simplex (Simp. Conv.)
$T_{vol_n} \rightarrow$ Tolerance to the normalized volume of the simplex
$* \rightarrow$ Parameter non used in the method