Penalty-Based Heuristic DIRECT Method for Constrained Global Optimization

M. Fernanda P. Costa[1(✉)], Ana Maria A. C. Rocha[2], and Edite M. G. P. Fernandes[2]

[1] Centre of Mathematics, University of Minho, 4710-057 Braga, Portugal
mfc@math.uminho.pt

[2] ALGORITMI Center, University of Minho, 4710-057 Braga, Portugal
{arocha,emgpf}@dps.uminho.pt

Abstract. This paper is concerned with an extension of the heuristic DIRECT method, presented in [8], to solve nonlinear constrained global optimization (CGO) problems. Using a penalty strategy based on a penalty auxiliary function, the CGO problem is transformed into a bound constrained problem. We have analyzed the performance of the proposed algorithm using fixed values of the penalty parameter, and we may conclude that the algorithm competes favourably with other DIRECT-type algorithms in the literature.

Keywords: Global optimization · DIRECT method · Heuristic · Penalty auxiliary function

1 Introduction

In this paper, we aim to find the global solution of a non-smooth and non-convex constrained optimization problem using a non-differentiable penalty function and the DIRECT method [1]. The constrained global optimization (CGO) problem has the form:

$$\begin{aligned} \min_{x\in\Omega} \; & f(x) \\ \text{subject to } \; & h(x)=0 \\ & g(x)\le 0, \end{aligned} \tag{1}$$

where $f:\mathbb{R}^n\to\mathbb{R}$, $h:\mathbb{R}^n\to\mathbb{R}^m$ and $g:\mathbb{R}^n\to\mathbb{R}^p$ are nonlinear continuous functions and $\Omega=\{x\in\mathbb{R}^n : -\infty < l_i \le x_i \le u_i < \infty, i=1,\ldots,n\}$. Denoting the feasible set of problem (1) by $\mathcal{F}=\{x\in\Omega : h(x)=0, g(x)\le 0\}$, we define a non-negative function

$$\theta(x)=\sum_{i=1}^{m}|h_j(x)|+\sum_{i=1}^{p}\max\{g_i(x),0\}, \tag{2}$$

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12251, pp. 538–551, 2020.
https://doi.org/10.1007/978-3-030-58808-3_39

where $\theta(x) = 0$ if $x \in \mathcal{F}$. Since convexity is not assumed, many local minima may exist in the feasible region, although we require only a global solution. For non-smooth problems, the derivative-free methods are the most appropriate. Deterministic and stochastic methods have been proposed to solve CGO problems [2,3]. Using deterministic methods, the convergence to a global optimal solution can be guaranteed and a solution with a required accuracy is obtained in a finite number of steps. On the other hand, stochastic methods are not guaranteed to find a global optimal solution although they are often able to find very good solutions after a (moderate) large number of steps. Stochastic convergence may be established using probability theory.

From the class of deterministic methods, the DIRECT method [1] has proven to be quite effective in converging to the global solution while avoiding to be trapped in a local solution, as far as bound constrained global optimization problems are concerned. The method has attracted considerable interest from the research community and several strategies have been incorporated into DIRECT, including the local search reinforcement [4,5], the improvement of the global search [6], new ideas for the selection of potentially optimal hyperrectangles [7,8] and new partition schemes [9–11].

The most popular methods to solve the problem (1) combine the objective function with a penalty term that aims to penalize constraint violation. Penalty functions within a DIRECT-type framework are proposed in [12–14]. An auxiliary function that combines in a special manner information on the objective and constraints is presented in [15]. Other techniques that involve the handling of the objective function and constraints violation separately can be found in [5,16,17].

The main contribution of this paper is the following. The two-phase heuristic DIRECT algorithm, proposed by the authors in [8], is extended to solve CGO problems using an auxiliary penalty function. The auxiliary function proposed in [15] is redefined to transform the CGO problem (1) into a bound constrained global optimization (BCGO) problem.

The paper is organized as follows. Section 2 briefly presents some ideas and the main steps of the DIRECT method. Section 3 describes a heuristic incorporated into the DIRECT algorithm to reduce the number of identified potentially optimal hyperrectangle, and the corresponding proposed extension to handle CGO problems, in particular, the use of a non-differentiable auxiliary penalty function. Finally, Sect. 4 contains the results of our preliminary numerical experiments and we conclude the paper with the Sect. 5.

2 DIRECT Method

The DIRECT (DIviding RECTangles) algorithm [1], originally proposed to solve BCGO problems of the form

$$\min_{x \in \Omega} f(x), \tag{3}$$

assumes that the objective function f is a continuous function, and iteratively produces finer and finer partitions of the hyperrectangles generated from Ω (see also [18]). The algorithm is a modification of the standard Lipschitzian approach, where f is assumed to satisfy the Lipschitz condition,

$$|f(x_1) - f(x_2)| \leq K\|x_1 - x_2\| \text{ for all } x_1, x_2 \in \Omega,$$

and the Lipschitz constant $K > 0$ is viewed as a weighting parameter that indicates how much emphasis to place on global versus local search. DIRECT is a deterministic method that does not require any analytical or numerical derivative information and searches (locally and globally) the feasible region Ω for hyperrectangles that are known as potentially optimal hyperrectangle (POH). These POH satisfy the two conditions established in the following definition:

Definition 1. *Given the partition $\{\mathcal{H}^i : i \in H\}$ of Ω, let ϵ be a positive constant, and let $f_{\min}$ be the current best function value. A hyperrectangle j is said to be potentially optimal if there exists some rate-of-change constant $\hat{K}^j > 0$ such that*

$$\begin{aligned} f(c_j) - \frac{\hat{K}^j}{2}\|u^j - l^j\| &\leq f(c_i) - \frac{\hat{K}^i}{2}\|u^i - l^i\|, \ \text{for all } i \in H \\ f(c_j) - \frac{\hat{K}^j}{2}\|u^j - l^j\| &\leq f_{\min} - \epsilon|f_{\min}| \end{aligned} \tag{4}$$

where c_j (resp. c_i) is the center and $\|u^j - l^j\|/2$ (resp. $\|u^i - l^i\|/2$) represents the size of hyperrectangle $j \in H$ (resp. i), and H is the set of indices of the hyperrectangles at the current iteration [1, 15].

The use of $\hat{K}^j$ in the definition intends to show that it is not the Lipschitz constant. The second condition in (4) aims to prevent the algorithm from identifying as POH the hyperrectangle with center that corresponds to $f_{\min}$. This way, small hyperrectangles where very small improvements may be obtained are skipped to be further divided.

The most important step in the DIRECT algorithm is the identification of POH since it determines the search along the feasible set. Each identified hyperrectangle is trisected along its longest sides and two new points in the hyperrectangle are sampled and remain center points of the other hyperrectangles (of the trisection).

A global search driven strategy would identify POH from the biggest hyperrectangles. On the other hand, a local search driven strategy would identify POH whose center point corresponds to $f_{\min}$. Good solutions are found rather quick but the hyperrectangle that contains the global solution may be missed if its center point has a bad function value. The main steps of the DIRECT algorithm are shown in Algorithm 1.

Algorithm 1. DIRECT algorithm

Require: η, f^*, $Nfe_{\max}$;
1: Set $Nfe = 0$;
2: **repeat**
3: Identification procedure for POH (Selection) according to Definition 1;
4: Selection procedure for division along dimensions (Sampling);
5: Division procedure;
6: Update index sets; Update Nfe;
7: **until** $Nfe \geq Nfe_{\max}$ or $|f_{\min} - f^*| \leq \eta \max\{1, |f^*|\}$

3 Heuristic DIRECT Method Based on Penalties

This section presents the extension of a heuristic DIRECT algorithm [8] to handle nonlinear equality and inequality constraints.

3.1 Heuristic DIRECT Method

Firstly, we briefly describe a heuristic that can be incorporated into the DIRECT algorithm [8] aiming

- to divide a promising search region into three subregions, so that the number of hyperrectangles that are candidate to be potentially optimal is reduced;
- to choose between a global search driven phase or a local search driven phase.

Since avoiding the identification of POH that were mostly divided can enhance the global search capabilities of DIRECT [6] and identifying POH that are close to the hyperrectangle which corresponds to $f_{\min}$ may improve the local search process, the heuristic incorporated into the DIRECT method divides the region of the hyperrectangles with least function values in each *size* group - denoted by *candidate* hyperrectangles - into three subregions.

Each subregion is defined by the indices based on *size* of the hyperrectangles. The larger the *size* the smaller the index. The leftmost subregion includes hyperrectangles whose indices are larger than $i_l = \lfloor 2/3 i_{\min} \rfloor$, where $i_{\min}$ is the index of the hyperrectangle that corresponds to $f_{\min}$. The rightmost subregion contains the hyperrectangles with indices that are smaller than $i_u = \lfloor 1/3 i_{\min} \rfloor$ and the middle subregion contains hyperrectangles with indices between i_l and i_u (including these limits).

To be able to guarantee convergence to the global solution while avoiding the stagnation in a local solution, the algorithm cycles between global and local search phases. It starts with a global driven search, where $G_{\max}$ iterations are performed using all *candidate* hyperrectangles from the rightmost subregion, 50% of the *candidate* hyperrectangles from the middle subregion (randomly selected) and 10% of the *candidate* hyperrectangles from the leftmost subregion (randomly selected). This choice of percentages is hereinafter denoted by (10/50/100)%. At each iteration, the set of POH are identified among these selected hyperrectangles. Afterwards, a local driven search is implemented for $L_{\max}$ iterations with the percentages of selected *candidate* hyperrectangles in the leftmost and rightmost subregions changed, denoted by (100/50/10)%. This cycling process is repeated until convergence.

3.2 Penalty Auxiliary Function

We now extend this heuristic DIRECT method to handle nonlinear equality and inequality constraints. We use an auxiliary function that takes into consideration the violation of inequality constraints by combining information of the objective and constraint functions [15]. This function penalizes any deviation of the function value at the center c_j of a hyperrectangle above the global optimal value f^*:

$$P(c_j) = \max\{f(c_j) - f^*, 0\} + \sum_{i=1}^{p} \mu_i \max\{g_i(c_j), 0\} \tag{5}$$

where μ_i are positive weighting coefficients. Note that when the hyperrectangle has a feasible center point c_j, the second term is zero, and when it is infeasible, the second term is positive and the first term only counts for the cases where $f(c_j)$ is above f^*. Since f^* is unknown in general, but satisfies $f^* \leq f_{\min} - \varepsilon$, for a small tolerance $\varepsilon > 0$, we redefine the following variant of the auxiliary function

$$P(x;\mu) = \max\{f(x) - (f_{\min} - \varepsilon), 0\} + \mu \left(\sum_{i=1}^{m} |h_i(x)| + \sum_{i=1}^{p} \max\{g_i(x), 0\} \right) \tag{6}$$

where $f_{\min}$ is the current best function value found so far among all feasible center points. Although different weights might prove to be useful for some problems, we consider only one constant weighting coefficient for all the constraints, and extend the penalized constraint violation term to the equality constraints, since in our formulation they are treated separately from the inequality constraints. We remark that if no feasible point has been found so far, the function $P(x;\mu)$ is reduced to the second term alone in (6).

The definition of POH (recall Definition 1 above) is now adapted to the strategy that aims to find a global minimum solution of the problem

$$\min_{x \in \Omega} P(x;\mu) \tag{7}$$

for a fixed $\mu > 0$ value, in the sense that the sequence of approximations $x^k_{\min}$ (resp. $f^k_{\min}$) converges to x^* (resp. f^*), the global optimal solution of problem (1), as k increases. In this context, the new algorithm searches (locally and globally) the feasible region Ω to identify hyperrectangles that are known as POH with respect to $P(x;\mu)$ and satisfy:

Definition 2. *Given the partition* $\{\mathcal{H}^i : i \in H\}$ *of* Ω, *let* $\epsilon > 0$ *and* $\mu > 0$ *be constants and let* $f_{\min}$ *be the current best function value among feasible center* *points. A hyperrectangle j is said to be potentially optimal with respect to $P(x;\mu)$ if there exists some rate-of-change constant $\hat{K}^j > 0$ such that*

$$\begin{aligned} P(c_j;\mu) - \frac{\hat{K}^j}{2}\|u^j - l^j\| &\le P(c_i;\mu) - \frac{\hat{K}^i}{2}\|u^i - l^i\|, \ \ \textit{for all } i \in H \\ P(c_j;\mu) - \frac{\hat{K}^j}{2}\|u^j - l^j\| &\le P_{\min} - \epsilon|P_{\min}| \end{aligned} \tag{8}$$

where $P_{\min}$ is the current best penalty function value and H is the set of indices of the selected candidate *hyperrectangles at the current iteration.*

The main steps of the proposed penalty-based heuristic DIRECT algorithm are presented in Algorithm 2.

Algorithm 2. Penalty-based heuristic DIRECT algorithm

Require: η_1, η_2, $G_{\max}$, $L_{\max}$, f^*, $Nfe_{\max}$;
Set $Nfe = 0$, $flag = G$, $it = 0$;
repeat
  Set $it = it + 1$;
  if $flag = G$ **then**
    Based on $i_{\min}$ and function P, randomly select the *candidate* hyperrectangles from the 3 subregions of indices based on the percentages (10/50/100)%;
  else
    Based on $i_{\min}$ and function P, randomly select the *candidate* hyperrectangles from the 3 subregions of indices based on the percentages (100/50/10)%;
  end if
  Identification procedure for POH according to Definition 2, among those selected *candidate* hyperrectangles (Selection);
  Selection procedure for division along dimensions (Sampling);
  Division procedure;
  Update index sets; Update Nfe;
  if $flag = L$ and $it \ge L_{max}$ **then**
    Set $flag = G$, $it = 0$;
  end if
  if $flag = G$ and $it \ge G_{max}$ **then**
    Set $flag = L$, $it = 0$;
  end if
until $Nfe \ge Nfe_{\max}$ or $(\theta(x_{\min}) \le \eta_1$ and $|f_{\min} - f^*| \le \eta_2 \max\{1, |f^*|\}$)

Unless otherwise stated, the stopping conditions for the algorithm are the following. We consider that a good approximate solution x^k, at iteration k, is found, if the conditions

$$\theta(x^k_{\min}) \le \eta_1 \ \text{ and } \ \frac{|f^k_{\min} - f^*|}{\max\{1, |f^*|\}} \le \eta_2 \tag{9}$$

are satisfied, for sufficiently small tolerances $\eta_1, \eta_2 > 0$, where $x^k_{\min}$ is the best computed solution to the problem, i.e., is the feasible center point of the hyperrectangle that has the least function value $f^k_{\min}$. However, if conditions (9) are not satisfied, the algorithm runs until a maximum number of function evaluations, $Nfe_{\max}$, is reached.

4 Numerical Experiments

In these preliminary numerical experiments, a set of seven benchmark problems with $n \leq 5$ is used. The MATLAB® (MATLAB is a registered trademark of the MathWorks, Inc.) programming language is used to code the algorithm and the tested problems. The parameter values for the algorithm are set as follows: ϵ = 1E-04, $G_{\max} = 10$, $L_{\max} = 10$, ε = 1E−06, η_1 = 1E-04, η_2 = 1E−04 and $Nfe_{\max}$ = 1E+05. Due to the random issue present in the algorithm, namely the selection of the *candidate* hyperrectangles, each problem is run five times. The reported results in the subsequent tables correspond to average values obtained in the five runs.

First, we consider two problems, one has 2 variables and 2 inequality constraints and the other 3 variables and 3 equality constraints, to show the effectiveness of the proposed strategy based on the penalty auxiliary function (6) when compared to the more usual L1 penalty-based technique.

Problem 1. (Problem 8 in [19])

$$\begin{aligned} &\min_{x \in \Omega} x_1^4 - 14x_1^2 + 24x_1 - x_2^2 \\ &\text{s. t. } x_2 - x_1^2 - 2x_1 \leq -2 \\ &\qquad -x_1 + x_2 \leq 8 \end{aligned}$$

with $\Omega = \{x \in \mathbb{R}^2 : -8 \leq x_1 \leq 10,\ 0 \leq x_2 \leq 10\}$ and $f^* = -118.70$.

Problem 2. (Problem 5 in [19])

$$\begin{aligned} &\min_{x \in \Omega} x_3 \\ &\text{s. t. } 30x_1 - 6x_1^2 - x_3 = -250 \\ &\qquad 20x_2 - 12x_2^2 - x_3 = -300 \\ &\qquad 0.5(x_1 + x_2)^2 - x_3 = -150 \end{aligned}$$

with $\Omega = \{x \in \mathbb{R}^3 : 0 \leq x_1 \leq 9.422,\ 0 \leq x_2 \leq 5.903,\ 0 \leq x_3 \leq 267.42\}$ and $f^* = 201.16$.

To analyze the gain in efficiency of Algorithm 2, we report in Table 1 the average values of f ($f_{\min}$), θ ($\theta(x_{\min})$), number of iterations (k) and number of function evaluations (Nfe) obtained after the 5 runs using the stopping conditions in (9) (or a maximum of 1E+05 function evaluations is reached). The standard deviation of the obtained f values ($St.D.$) is also reported. From the results in Table 1 we can conclude that the algorithm with the penalty (6) gives significantly better results than with the penalty L1. Similarly, from Table 2 we conclude that penalty (6) performs much better compared to L1. The algorithm

Table 1. Comparison between penalty functions, when solving Problem 1.

Algorithm 2	μ	$f_{\min}$	$St.D.$	$\theta(x_{\min})$	k	Nfe
Penalty (6)	0.5	−118.691829	2.26E−03	0.00E+00	109	2175
	1	−118.692153	2.01E−03	0.00E+00	86	1457
	10	−118.688994	2.65E−04	0.00E+00	44	526
	100	−118.688657	1.69E−04	0.00E+00	77	1121
L1 penalty	0.5	(−217)	(6.15E−05)	(4.5E+00)	(4848)	>1E+05
	1	(−217)	(6.15E−05)	(4.5E+00)	(4852)	>1E+05
	10	(−215)	(5.02E−05)	(4.0E+00)	(4828)	>1E+05
	100	−118.689247	6.48E−04	0.00E+00	72	865

In parentheses, achieved values when the algorithm stops due to $Nfe >$ 1E+05.

Table 2. Comparison between penalty functions, when solving Problem 2.

Algorithm 2	μ	$f_{\min}$	$St.D.$	$\theta(x_{\min})$	k	Nfe
Penalty (6)	0.5	201.159343	0.00E+00	7.83E−05	43	577
	1	201.159343	0.00E+00	7.83E−05	43	543
	10	201.159343	0.00E+00	7.83E−05	45	624
	100	201.159343	0.00E+00	7.83E−05	41	531
Penalty L1	0.5	(201)	(0.00E+00)	(7.6E−04)	(8803)	>1E+05
	1	(201)	(0.00E+00)	(1.4E−04)	(7612)	>1E+05
	10	201.159343	0.00E+00	7.83E−05	320	5864
	100	201.159343	0.00E+00	7.83E−05	45	577

In parentheses, achieved values when the algorithm stops due to $Nfe >$ 1E+05.

with the penalty L1 works better with the larger values of the weighing parameter while the performance of the algorithm with penalty (6) is not too much affected by the value of μ.

In Table 3 we compare the results obtained by Algorithm 2 based on the penalty auxiliary function (6) for two values of the weighting parameter (that provide the best results among the four tested) with those obtained by previous DIRECT-type strategies that rely on the filter methodology to reduce both the constraint violation and objective function [8,17]. The results are also compared to those obtained by variants DIRECT-GL and DIRECT-GLce reported in [14]. We note that the reported $f_{\min}$, $\theta(x_{\min})$, k and Nfe selected from [8] correspond also to average values, while the values from the other papers in comparison correspond to just a single solution (one run of deterministic methods). A slight gain in efficiency of the proposed penalty-based heuristic DIRECT algorithm has been detected.

To analyze the performance of the Algorithm 2 when compared to the strategy proposed in [15] and two filter-based DIRECT algorithms (in [8,17]), we consider the problem Gomez #3 (available in [15]):

Table 3. Comparative results for Problems 1 and 2.

	μ	$f_{\min}$	$\theta(x_{\min})$	k	Nfe	f^*
Problem 1						
Algorithm 2 (penalty (6))	10	−118.688994	0.00E+00	44	526	−118.70
	100	−118.688657	0.00E+00	77	1121	
DIRECT-type[a] [8]		−118.700976	0.00E+00	19	823	
DIRECT-type[b] [8]		−118.700976	0.00E+00	19	797	
DIRECT-type[c] [8]		−118.692210	0.00E+00	23	689	
filter-based DIRECT [17]		−118.700976	0.00E+00	23	881	
DIRECT-GLc in [14]		−118.6892	–	–	1197	
DIRECT-GLce in [14]		−118.6898	–	–	1947	
Problem 2						
Algorithm 2 (penalty (6))	1	201.159343	7.83E−05	43	543	201.16
	100	201.159343	7.83E−05	41	531	
DIRECT-type[a] [8]		201.159343	7.83E−05	30	1015	
DIRECT-type[b] [8]		201.159343	7.83E−05	30	883	
DIRECT-type[c] [8]		201.159343	7.83E−05	30	769	
filter-based DIRECT [17]		201.159343	7.83E−05	30	1009	
DIRECT-GLc in [14]		201.1593	–	–	819	
DIRECT-GLce in [14]		201.1593	–	–	819	

[a]with filter; [b]with filter and upper bounds on f and θ;
[c]with filter and upper bounds on f and θ as well as a heuristic.
"–" information not available.

Problem 3.

$$\begin{array}{l}\min\limits_{x\in\Omega} \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right) x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2 \\ \text{s. t. } -\sin(4\pi x_1) + 2\sin^2(2\pi x_2) \leq 0\end{array}$$

with $\Omega = \{x \in \mathbb{R}^2 : -1 \leq x_i \leq 1,\ i = 1, 2\}$.

Table 4 compares the performance of the tested algorithms. Our Algorithm 2 was tested with four different values of the fixed weighting parameter. When solving the Problem 3, our algorithm reports a considerable sensitivity to the selected μ value, with a better performance achieved when small values are used.

The following problem, known as T1 is tested with 3 different values of n.

Problem 4.

$$\begin{array}{l}\min\limits_{x\in\Omega} \sum_{i=1}^n x_i \\ \text{s. t. } \sum_{i=1}^n x_i^2 \leq 6\end{array}$$

with $\Omega = \{x \in \mathbb{R}^n : -1 \leq x_i \leq 1,\ i = 1, \ldots, n\}$.

Table 4. Comparison results when solving Problem 3.

	μ	$f_{\min}$	$\theta(x_{\min})$	k	Nfe	f^*
Algorithm 2 (penalty (6))	0.5	−0.971021	2.45E−05	51	606	−0.9711
	1	−0.971018	1.34E−05	74	983	
	10	−0.971018	1.34E−05	455	12952	
	100	(−0.97)	(4.4E−05)	(2625)	>1E+05	
DIRECT-type[a] [8]		−0.971006	6.00E−05	17	615	
DIRECT-type[b] [8]		−0.971006	6.00E−05	17	683	
DIRECT-type[c] [8]		−0.971041	3.17E−05	18	555	
filter-based DIRECT [17]		–	–	18	733	
DIRECT in [15]		–	–	–	513	

In parentheses, the achieved values when the algorithm stops due to $Nfe > $ 1E+05.
[a]with filter; [b]with filter and upper bounds on f and θ;
[c]with filter and upper bounds on f and θ as well as a heuristic.
"–" information not available.

The results obtained by Algorithm 2, and those in [14] (variants DIRECT-GLc and DIRECT-GLce), as well as the results obtained by the variant DIRECT-GL and the original DIRECT (when they are implemented in a penalty-based strategy with penalty function L1) are shown in Table 5 for comparison. Our algorithm with the penalty (6) works much better with the smaller values of the fixed weighting parameter and those results outperform in general the other results in comparison, for the same solution quality accuracy, as far as function evaluations are concerned.

Finally, the last 3 problems, known as g04, g06 and g08 in [16], have inequality constraints.

Problem 5.

$$\begin{aligned}
&\min_{x\in\Omega}\ 5.3578547x_3^2+0.8356891x_1x_5+37.293239x_1-40792.141\\
&\text{s. t. } 0\le 85.334407+0.0056858x_2x_5+0.0006262x_1x_4-0.0022053x_3x_5\le 92\\
&\qquad 90\le 80.51249+0.0071317x_2x_5+0.0029955x_1x_2+0.0021813x_3^2\le 110\\
&\qquad 20\le 9.300961+0.0047026x_3x_5+0.0012547x_1x_3+0.0019085x_3x_4\le 25
\end{aligned}$$

with $\Omega=\{x\in\mathbb{R}^5: 78\le x_1\le 102,\ 33\le x_2\le 45,\ 27\le x_i\le 45, i=3,4,5\}$.

Problem 6.

$$\begin{aligned}
&\min_{x\in\Omega}\ (x_1-10)^3+(x_2-20)^3\\
&\text{s. t. } -(x_1-5)^2-(x_2-5)^2\le -100\\
&\qquad (x_1-6)^2+(x_2-5)^2\le 82.81
\end{aligned}$$

with $\Omega=\{x\in\mathbb{R}^2: 13\le x_1\le 100,\ 0\le x_2\le 100\}$.

Table 5. Comparison results when solving Problem 4.

	μ	$f_{\min}$	$\theta(x_{\min})$	k	Nfe	f^*
$n = 2$						
Algorithm 2 (penalty (6))	0.5	−3.464079	3.60E−05	26	383	−3.4641
	1	−3.464052	3.30E−05	26	370	
	10	−3.464106	9.29E−05	40	723	
	100	−3.464106	7.68E−05	85	3927	
DIRECT-type[a] [8]		−3.464106	9.29E−05	14	1395	
DIRECT-type[b] [8]		−3.464106	9.29E−05	14	893	
DIRECT-type[c] [8]		−3.464106	5.72E−05	13	335	
DIRECT-L1 in [14]	10	–	–	–	3345	
	100	–	–	–	8229	
DIRECT-GL-L1 in [14]	10	–	–	–	1221	
	100	–	–	–	1921	
DIRECT-GLc in [14]		–	–	–	1373	
DIRECT-GLce in [14]		–	–	–	2933	
$n = 3$						
Algorithm 2 (penalty (6))	0.5	−4.242687	7.25E−05	266	29187	−4.2426
	1	−4.242443	4.38E−05	104	6989	
	10	−4.242443	0.00E+00	110	8577	
	100[d]	−4.242443	2.30E−05	260	85472	
DIRECT-type[a] [8]		−4.242443	0.00E+00	28	16885	
DIRECT-type[b] [8]		−4.242443	0.00E+00	35	37977	
DIRECT-type[c] [8]		−4.242443	9.17E−05	29	3233	
DIRECT-L1 in [14]	10	–	–	–	66137	
	100	–	–	–	>1E+06	
DIRECT-GL-L1 in [14]	10	–	–	–	75105	
	100	–	–	–	16625	
DIRECT-GLc in [14]		–	–	–	26643	
DIRECT-GLce in [14]		–	–	–	8297	
$n = 4$						
Algorithm 2 (penalty (6))	0.5	−4.898440	0.00E+00	74	9514	−4.899
	1	−4.898440	0.00E+00	62	6201	
	10[e]	−4.898440	3.42E−05	133	54981	
	100[d]	−4.898440	5.80E−05	98	31440	
DIRECT-type[a] [8]		−4.898847	0.00E+00	42	151753	
DIRECT-type[b] [8]		−4.898847	3.42E−05	39	78859	
DIRECT-type[c] [8]		−4.898440	3.30E−05	51	36219	
DIRECT-L1 in [14]	10	–	–	–	127087	
	100	–	–	–	>1E+06	
DIRECT-GL-L1 in [14]	10	–	–	–	180383	
	100	–	–	–	189595	
DIRECT-GLc in [14]		–	–	–	192951	
DIRECT-GLce in [14]		–	–	–	47431	

[a] with filter; [b] with filter and upper bounds on f and θ;
[c] with filter and upper bounds on f and θ as well as a heuristic.
[d] 80% successful runs; [e] 60% successful runs.
"–" information not available.

Problem 7.

$$\min_{x \in \Omega} -\frac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1+x_2)}$$
$$\text{s. t. } x_1^2 - x_2 + 1 \leq 0$$
$$1 - x_1 - (x_2-4)^2 \leq 0$$

with $\Omega = \{x \in \mathbb{R}^2 : 0 \leq x_i \leq 10, i = 1, 2\}$.

Table 6. Comparison results when solving Problems 5, 6 and 7.

	μ	$f_{\min}$	$\theta(x_{\min})$	k	Nfe	f^*
Problem 5[a]						
Algorithm 2 (penalty (6))	0.5[b]	−30665.2339	9.99E−05	387	36277	−30665.53867
	1[c]	−30665.4237	9.99E−05	377	40331	
	10	−30665.2450	9.96E−05	132	5119	
	100	−30665.2329	9.79E−05	133	5247	
	1000	−30665.2329	9.79E−05	135	5746	
DIRECT-GL-L1 in [14]	1000	–	–	–	1799	
DIRECT-GLc in [14]		–	–	–	5907	
DIRECT-GLce in [14]		−30663.5708	–	–	21355	
eDIRECT-C in [16]		−30665.5385	–	–	65	
Problem 6[a]						
Algorithm 2 (penalty (6))	0.5	−6961.9092	6.86E−05	120	2815	−6961.81387558
	1	−6961.8763	4.80E−05	118	2699	
	10	−6961.8088	2.18E−05	114	2758	
	100	−6961.7868	1.66E−05	121	2939	
	1000	−6961.8150	2.77E−05	186	4941	
DIRECT-GL-L1 in [14]	1000[d]	–	–	–	289	
DIRECT-GLc in [14]		–	–	–	3461	
DIRECT-GLce in [14]		−6961.1798	–	–	6017	
eDIRECT-C in [16]		−6961.8137	–	–	35	
Problem 7						
Algorithm 2 (penalty (6))	0.5	−0.095825	0.00E+00	18	174	−0.095825
	1	−0.095825	0.00E+00	16	158	
	10	−0.095825	0.00E+00	16	152	
	100	−0.095825	0.00E+00	16	164	
	1000	−0.095825	0.00E+00	15	153	
DIRECT-GL-L1 in [14]	1000	–	–	–	471	
DIRECT-GLc in [14]		–	–	–	471	
DIRECT-GLce in [14]		−0.0958	–	–	1507	
eDIRECT-C in [16]		−0.0958	–	–	154	

[a] results for η_2 = 1E−05; [b] 20% successful runs;
[c] 80% successful runs; [d] final solution outside the feasible region.
"–" information not available.

The results obtained by our Algorithm 2 for five values of μ are compared to those of the variants DIRECT-GL-L1, DIRECT-GLc and DIRECT-GLce in [14], and eDIRECT-C from [16]. We remark that eDIRECT-C incorporates a local minimization search (MATLAB `fmincon`). From the results in Table 6, we may conclude that contrary to Problem 5 which presents significantly better results with the larger fixed μ values, the other two problems report reasonable good performances with all the tested μ values, competing with the other algorithms is comparison.

5 Conclusions

In this paper, we present an extension of the heuristic DIRECT method (available in [8]) to solve nonlinear CGO problems. The herein proposed extension transforms the CGO problem (1) into a BCGO one, using a penalty strategy based on the penalty auxiliary function (6). We have analyzed the performance of the penalty-based heuristic DIRECT algorithm for a set of fixed penalty parameter values, using well-known benchmark CGO problems. Neither too small nor too large parameter values (1, 10 and 100) have produced results that show the robustness and efficiency of proposed algorithm hereby competing favourably with other available DIRECT-type algorithms.

Although, for now, we have considered a fixed value for the parameter ε (in the definition of the penalty (6)), we feel that a sequence of decreased values may further improve the efficiency of the algorithm. This will be an issue for future research.

Acknowledgments. The authors wish to thank two anonymous referees for their comments and suggestions to improve the paper.

This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020, UIDB/00013/2020 and UIDP/00013/2020 of CMAT-UM.

References

1. Jones, D.R., Perttunen, C.D., Stuckman, B.E.: Lipschitzian optimization without the Lipschitz constant. J. Optim. Theory Appl. **79**(1), 157–181 (1993). Based on exact penalty functions
2. Hendrix, E.M.T., G.-Tóth, B.: Introduction to Nonlinear and Global Optimization. Optimization and its Applications. Springer, New York (2010). https://doi.org/10.1007/978-0-387-88670-1
3. Sergeyev, Y.D., Kvasov, D.E.: Deterministic Global Optimization. SO. Springer, New York (2017). https://doi.org/10.1007/978-1-4939-7199-2
4. Gablonsky, J.M., Kelley, C.T.: A locally-biased form of the DIRECT algorithm. J. Glob. Optim. **21**(1), 27–37 (2001)
5. Di Pillo, G., Liuzzi, G., Lucidi, S., Piccialli, V., Rinaldi, F.: A DIRECT-type approach for derivative-free constrained global optimization. Comput. Optim. Appl. **65**(2), 361–397 (2016). https://doi.org/10.1007/s10589-016-9876-3
6. Paulavičius, R., Sergeyev, Y.D., Kvasov, D.E., Žilinskas, J.: Globally-biased DISIMPL algorithm for expensive global optimization. J. Glob. Optim. **59**(2–3), 545–567 (2014)

7. Stripinis, L., Paulavičius, R., Žilinskas, J.: Improved scheme for selection of potentially optimal hyper-rectangles in DIRECT. Optim. Lett. **12**(7), 1699–1712 (2018)
8. Costa, M.F.P., Fernandes, E.M.G.P., Rocha, A.M.A.C.: A two-phase heuristic coupled DIRECT method for bound constrained global optimization. In: EUROGEN 2019 Conference Proceedings, 8 p. Guimarães, Portugal (2019)
9. Paulavičius, R., Žilinskas. J.: Simplicial Lipschitz optimization without the Lipschitz constant. J. Glob. Optim. **59**, 23–40 (2014)
10. Liu, H., Xu, S., Wang, X., Wu, J., Song, Y.: A global optimization algorithm for simulation-based problems via the extended DIRECT scheme. Eng. Optim. **47**(11), 1441–1458 (2015)
11. Paulavičius, R., Sergeyev, Y.D., Kvasov, D.E., Žilinskas, J.: Globally-biased BIRECT algorithm with local accelerators for expensive global optimization. Expert. Syst. Appl. **144**, 113052 (2020)
12. Di Pillo, G., Lucidi, S., Rinaldi, F.: An approach to constrained global optimization based on exact penalty functions. J. Glob. Optim. **54**(2), 251–260 (2012)
13. Di Pillo, G., Lucidi, S., Rinaldi, F.: A derivative-free algorithm for constrained global optimization based on exact penalty functions. J. Optim. Theory Appl. **164**, 862–882 (2015)
14. Stripinis, L., Paulavičius, R., Žilinskas, J.: Penalty functions and two-step selection procedure based DIRECT-type algorithm for constrained global optimization. Struct. Multidisc. Optim. **59**, 2155–2175 (2019)
15. Jones, D.R.: The DIRECT global optimization algorithm. In: Floudas, C., Pardalos, P. (eds.) Encyclopedia of Optimization, pp. 431–440. Kluwer Academic Publisher, Boston (2001)
16. Liu, H., Xu, S., Chen, X., Wang, X., Ma, Q.: Constrained global optimization via a DIRECT-type constraint-handling technique and an adaptive metamodeling strategy. Struct. Multi. Optim. **55**(1), 155–177 (2016). https://doi.org/10.1007/s00158-016-1482-6
17. Costa, M.F.P., Rocha, A.M.A.C., Fernandes, E.M.G.P.: Filter-based DIRECT method for constrained global optimization. J. Glob. Optim. **71**(3), 517–536 (2017). https://doi.org/10.1007/s10898-017-0596-8
18. Gablonsky, J.M.: DIRECT version 2.0 user guide. Technical report CRSC-TR-01-08, Center for Research in Scientific Computation, North Carolina State University (2001)
19. Birgin, E.G., Floudas, C.A., Martínez, J.M.: Global minimization using an Augmented Lagrangian method with variable lower-level constraints. Technical report MCDO121206, Department of Computer Science IME-USP, University of São Paulo (2007). http://www.ime.usp.br/~egbirgin/