Comparison of Penalty Functions on a Penalty Approach to Mixed-Integer Optimization

Rogério B. Francisco*, M. Fernanda P. Costa†, Ana Maria A.C. Rocha** and Edite M.G.P. Fernandes**

*School of Technology and Management Felgueiras, 4610-156 Felgueiras, Portugal
†Centre of Mathematics, University of Minho, 4710-057 Braga, Portugal
**Algoritmi Research Centre, University of Minho, 4710-057 Braga, Portugal

Abstract. In this paper, we present a comparative study involving several penalty functions that can be used in a penalty approach for globally solving bound mixed-integer nonlinear programming (bMIMLP) problems. The penalty approach relies on a continuous reformulation of the bMINLP problem by adding a particular penalty term to the objective function. A penalty function based on the 'erf' function is proposed. The continuous nonlinear optimization problems are sequentially solved by the population-based firefly algorithm. Preliminary numerical experiments are carried out in order to analyze the quality of the produced solutions, when compared with other penalty functions available in the literature.

Keywords: Mixed-integer, Penalty function, Firefly algorithm
PACS: 02.60.Pn

INTRODUCTION

This paper aims to compare the performance of several penalty functions on an exact penalty approach for globally solving bMINLP problems. The presented penalty approach seeks for a global solution of the bMINLP problem by replacing it by a sequence of continuous nonlinear programming (NLP) problems with only continuous variables. The mathematical formulation of the problem to be addressed has the form:

$$\begin{array}{rl} \min & f(x) \\ \text{subject to} & x \in \Omega \subset \mathbb{R}^n \\ & x_i \in \mathbb{R} \text{ for } i = 1,\ldots,n_c, \quad x_j \in \mathbb{Z} \text{ for } j = n_c+1,\ldots,n \end{array} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuous function and Ω is a compact convex set. In this paper, the set $\Omega = \{x \in \mathbb{R}^n : l_i \leq x_i \leq u_i, i = 1,\ldots,n\}$ where l and u are the vectors of the lower and upper bounds respectively. n_c denotes the number of continuous variables, $n_i = n - n_c$ gives the number of integer variables. The feasible set W of bMINLP problem (1) is defined by $W = \{x \in \Omega \subset \mathbb{R}^n : x_j \in \mathbb{Z} \text{ for } j = n_c+1,\ldots,n\}$.

We consider the following continuous reformulation of the bMINLP problem (1), which comes out by relaxing all integer variables to continuous ones and adding a particular penalty term to the objective function:

$$\begin{array}{rll} \min & \Psi(x;\varepsilon) & \equiv f(x) + \phi(x;\varepsilon) \\ \text{subject to} & x \in \Omega & \text{and } x_i \in \mathbb{R} \text{ for } i = 1,\ldots,n \end{array} \tag{2}$$

where $\phi(x;\varepsilon)$ is the penalty function. In [1] it is shown that, the problems (1) and (2) are equivalent, for any $\varepsilon \in (0,\bar{\varepsilon}]$, in the sense that they have the same global minimizers. The penalty function in (2) is termed 'exact' since the two problems have the same global minimizers for a sufficiently small value of the penalty parameter ε. In this work we use a similar exact penalty algorithm as proposed in [1], which combines a global optimization technique for solving the continuous reformulation for a given value of the penalty parameter ε and an automatic updating of ε occurring a finite number of times. Problem (2) parameterized by ε is globally solved by a simple and stochastic population-based algorithm, known as firefly algorithm (FA) [2, 3]. We present a new penalty function based on the 'erf' function, and analyze its performance when compared with other penalty function available from the literature. Some preliminary results are presented with a benchmark set of mixed-integer nonlinear programming problems.

The remainder of this paper is as follows. First, five penalty terms known from the literature and the proposed 'erf' penalty function are presented. Second, the exact penalty algorithm and a brief description of the FA are provided. Finally, some numerical experiments and conclusions are reported.

PENALTY FUNCTIONS ON A PENALTY APPROACH

Exact penalty approaches have been used to solve general nonlinear integer programming problems [1, 4, 5, 6, 7]. From the class of penalty functions that can be used in this penalty approach, the two most used are:

$$\phi(x;\varepsilon) = \sum_{j\in J} \min_{\substack{l_j \le d_i \le u_j \\ d_i \in \mathbb{Z}}} \left\{ \log\left[|x_j - d_i| + \varepsilon\right] \right\} \tag{3}$$

$$\phi(x;\varepsilon) = \frac{1}{\varepsilon} \sum_{j\in J} \min_{\substack{l_j \le d_i \le u_j \\ d_i \in \mathbb{Z}}} \left\{ \left[|x_j - d_i| + \varepsilon\right]^p \right\},\ 0 < p < 1 \tag{4}$$

where the index set J is defined by $J = \{n_c + 1, \ldots, n\}$. Other penalty terms that may be considered are simple adaptations of those reported in [1] for the zero-one programming problem:

$$\phi(x;\varepsilon) \quad = \sum_{j\in J} \min_{\substack{l_j \le d_i \le u_j \\ d_i \in \mathbb{Z}}} \left\{ -\left[|x_j - d_i| + \varepsilon\right]^{-q} \right\},\ q > 0 \tag{5}$$

$$\phi(x;\varepsilon) = \frac{1}{\varepsilon} \sum_{j\in J} \min_{\substack{l_j \le d_i \le u_j \\ d_i \in \mathbb{Z}}} \left\{ 1 - \exp\left(-\rho |x_j - d_i|\right) \right\},\ \rho > 0 \tag{6}$$

$$\phi(x;\varepsilon) = \frac{1}{\varepsilon} \sum_{j\in J} \min_{\substack{l_j \le d_i \le u_j \\ d_i \in \mathbb{Z}}} \left\{ \left[1 + \exp\left(-\rho |x_j - d_i|\right)\right]^{-1} \right\},\ \rho > 0. \tag{7}$$

The new herein proposed penalty function based on the continuous 'erf' function for solving the bMINLP problem is the following. We note that the function $\mathrm{erf}(\cdot)$ is differentiable and strictly increasing on the set Ω. The penalty term takes the form

$$\phi(x;\varepsilon) = \frac{1}{\varepsilon} \sum_{j\in J} \min_{\substack{l_j \le d_i \le u_j \\ d_i \in \mathbb{Z}}} \left\{ \mathrm{erf}\left(|x_j - d_i| + \varepsilon \right) \right\}. \tag{8}$$

The exact penalty algorithm is composed by an outer cycle where, at each iteration k, the continuous reformulation problem (2) is solved for a fixed value of the penalty parameter $\varepsilon^{(k)}$. See Algorithm 1. At each iteration k, an approximate global minimizer $x^{(k)}$ of problem (2) is computed. Whenever the computed $x^{(k)}$ is not feasible for problem (1) and the condition

$$\psi(x^{(k)};\varepsilon^{(k)}) - \psi(z^{(k)};\varepsilon^{(k)}) \le \varepsilon^{(k)} \mathscr{L} \|x^{(k)} - z^{(k)}\|$$

is verified, where $z^{(k)} \in W$, the penalty parameter $\varepsilon^{(k)}$ is decreased and the tolerance for solution quality $\delta^{(k)}$ remains unchanged. Otherwise, $\varepsilon^{(k)}$ remains unchanged and $\delta^{(k)}$ is allowed to decrease. The point $z^{(k)}$ results from rounding $x_j^{(k)}, j \in J$ to the nearest integer. The algorithm terminates when the number of iterations, k, exceeds a given threshold value $k_{\max}$. This stopping criterium allows us to analyze the quality of the solutions produced by the algorithm. However, other stopping criteria may be used. Namely, if the optimal value is known, criteria based on the feasibility of $x^{(k)}$ and on the proximity of the objective function value to the known minimum may be used.

Our proposal for finding an approximate global minimizer of each problem (2), is based on the stochastic population-based FA. The FA is a bio-inspired population-based metaheuristic developed to solve global optimization problems with simple bounds. It is inspired by the flashing behavior of fireflies at night. The FA was developed by [2, 3] and is based on the following three main rules: a) it is assumed that all fireflies are unisex, meaning that they will be attracted to each other regardless of their sex; b) the attractiveness is proportional to their brightness but decrease as the distance increases. In the case of no existence of no brighter firefly, the fireflies will move randomly; c) the brightness of a firefly is determined from the encoded objective function to be optimized.

In the original version of FA, the main ideas to construct FA are related with the brightness emitted by each firefly and the degree of attractiveness that is generated between two fireflies. The movement of a firefly i towards another brighter firefly j is determined by:

$$x^i = x^i + \beta(x^j - x^i) + \alpha\varepsilon^i, \tag{9}$$

where x^i and x^j denote the location of fireflies i and j in the search space Ω, respectively. In (9), the second term is due to the attraction and the third term is due to randomization with $\alpha \in (0,1)$ being the randomization

Algorithm 1 Exact penalty algorithm

Require: $k_{\max}$, $\varepsilon^{(1)} > 0$, $\delta^{(1)} > 0$, $\mathscr{L} > 0$, $\sigma \in (0,1)$

Set $k = 1$
Randomly generate $x^{(0)} \in \Omega$
while $k \leq k_{\max}$ **do**
  Given $x^{(k-1)}$, compute an approximate global minimizer $x^{(k)}$ of problem (2) such that

$$\psi(x^{(k)};\varepsilon^{(k)}) \leq \psi(x;\varepsilon^{(k)}) + \delta^{(k)}, \quad \text{for all } x \in \Omega$$

  if $x^{(k)} \notin W$ and $\psi(x^{(k)};\varepsilon^{(k)}) - \psi(z^{(k)};\varepsilon^{(k)}) \leq \varepsilon^{(k)}\mathscr{L}\|x^{(k)} - z^{(k)}\|$ **then**
    Set $\varepsilon^{(k+1)} = \sigma\varepsilon^{(k)}$, $\delta^{(k+1)} = \delta^{(k)}$
  else
    Set $\varepsilon^{(k+1)} = \varepsilon^{(k)}$, $\delta^{(k+1)} = \sigma\delta^{(k)}$
  end if
  Set $k = k+1$
end while

parameter. Here, $\varepsilon^i = \mathrm{L}(0,1)\sigma^i/2$ where $\mathrm{L}(0,1)$ is a random number from the standard Lévy distribution and $\sigma^i = \left(|x_1^i - x_1^1|, \ldots, |x_n^i - x_n^1|\right)^T$ is a vector that gives the variation of x^i relative to the position of the brightest firefly, x^1. The parameter $\beta = \beta_0 \exp\left(-\gamma\|x^i - x^j\|^2\right)$ gives the attractiveness of a firefly i and varies with the brightness seen by adjacent firefly j and the distance between themselves. β_0 is the attraction parameter when the distance is zero. The parameter $\gamma \geq 0$ characterizes the variation of the attractiveness, and is crucial to speed the convergence of the algorithm.

NUMERICAL EXPERIMENTS

The numerical experiments were carried out on a PC Intel Core 2 Duo Processor E7500 with 2.9GHz and 4Gb of memory RAM. The algorithm was coded in Matlab Version 8.1 (R2013a). Eighteen bMINLP problems are used for the comparison of the six penalties. This comparison is based on the solution quality which is measured in terms of the difference between the best obtained result (out of 10 independent runs), f_{best}, and the known optimal value, f^*. Algorithm 1 is terminated when the number of iterations $k_{\max} = 20$. The parameters in the algorithm are set as follows: $m = 5n$, $\delta^{(1)} = 1e^{-5}$, $\varepsilon^{(1)} = 10$, $\mathscr{L} = 10$, $\sigma = 0.1$, $p = 0.5$, $q = 1$, $\rho = 1$. FA is allowed to run for 100 iterations, and $\beta_0 = 1$, α and γ are reduced as a function of the iteration counter until they reach 0.001, starting from 0.5 and 10, respectively.

From the results of Table 1 we may conclude that (considering 'wins' and 'ties') penalties (4) and (8) have similar performances with the best solutions in 72% and 67% of the tested problems respectively. The quality of the solutions produced by penalty (7) is better than or equal to the other cases in 50% of the problems and penalty (3) wins and produces ties on 33% of the problems. Penalties (5) and (6) perform poorly with only 22% of 'wins' and 'ties'.

Figure 1 displays the plots of the best four penalties using two values of ε: 1 (dotted lines) and 0.25 (solid lines). To identify (3), we use lines with the marker '$\circ$', to identify (4) the lines have a marker '$\square$', to identify (8) the lines are marked with '$\triangleright$', and to identify (7) the lines are marked with '$+$'. We note that penalties (4) and (8) have similar behavior as a function of ε and $|x - d_i|$.

CONCLUSIONS

This paper presents a practical comparison between penalty functions when used in an exact penalty approach to globally solve bMINLP problems. A new penalty term based on the 'erf' function is presented. In order to assess the performance of the proposed penalty algorithm, eighteen bMINLP problems are used. The numerical experiments carried out to compare the quality of the produced solutions show that the penalty function (8) is competitive when compared with the other tested five penalty terms.

TABLE 1. Comparison of $|f_{\text{best}} - f^*|$

Prob.	n	n_i	penalty (3)	penalty (4)	penalty (5)	penalty (6)	penalty (7)	penalty (8)
ACK_5	5	5	2.344E-02	1.492E-07	8.751E-01	3.931E-06	8.882E-16	1.096E-05
ACK_10	10	10	3.243E+00	1.609E+00	3.252E+00	2.630E+00	2.971E-01	1.651E+00
AP	2	1	8.607E-05	8.604E-05	3.203E-02	8.607E-05	8.605E-05	8.604E-05
Bea	2	1	3.890E-17	5.916E-20	4.776E-02	1.737E-09	2.444E-18	1.056E-17
BL	2	2	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
BF1	2	2	0.000E+00	0.000E+00	8.062E-01	3.124E-13	0.000E+00	0.000E+00
Buk	2	2	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
DA	2	2	4.829E-01	4.817E-01	4.856E-01	4.842E-01	4.823E-01	4.820E-01
DP_2	2	1	1.473E-16	1.142E-17	3.282E-02	4.548E-11	2.691E-20	2.936E-18
DP_4	4	1	1.194E-16	9.621E-17	8.693E-01	7.119E-07	3.256E-16	7.642E-17
Him	2	2	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
LM2_5	5	5	1.532E-05	1.500E-32	1.988E-01	1.231E-14	1.269E-31	1.500E-32
LM2_10	10	10	1.914E-04	1.500E-32	5.399E-01	7.891E-14	2.709E-30	1.500E-32
NF2	4	4	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
RG_5	5	5	2.077E-02	0.000E+00	4.732E-01	9.628E-12	0.000E+00	0.000E+00
RG_10	10	10	2.986E+00	0.000E+00	3.034E+00	4.014E+00	0.000E+00	0.000E+00
S10	4	4	9.095E-04	4.384E-03	5.058E-01	4.384E-03	4.384E-03	4.384E-03
SS_5	5	5	7.303E-05	1.277E-32	7.741E-01	2.123E-13	5.551E-30	1.530E-33

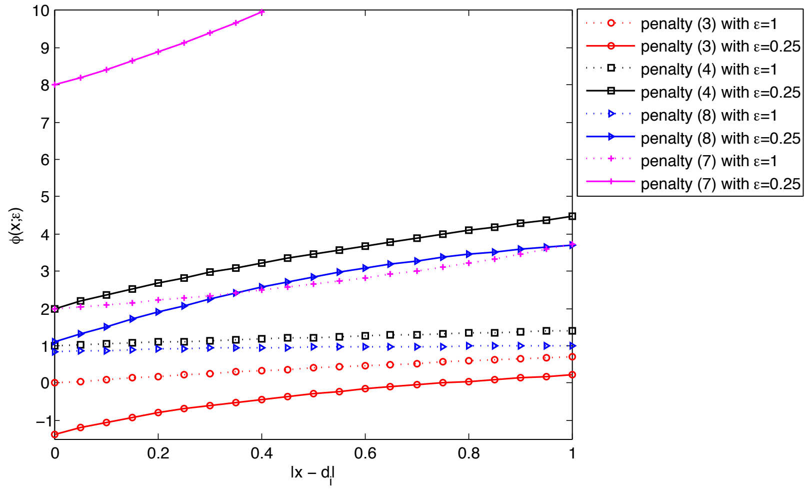

FIGURE 1. Plots of penalties (3), (4), (8) and (7)

ACKNOWLEDGMENTS

The authors would like to thank the financial support from FCT (Fundação para a Ciência e Tecnologia, Portugal) in the scope of the projects: PEst-OE/MAT/UI0013/2014 and PEst-UID/CEC/00319/2013.

REFERENCES

1. S. Lucidi, F. Rinaldi, *Journal of Optimization Theory and Applications*, **145(3)**, 479–488 (2010).
2. X.-S. Yang, SAGA 2009, O. Watanabe, T. Zeugmann (Eds.), Lect. Notes Comput. Sc., **5792**, 169–178 (2009).
3. X.-S. Yang, *Int. J. Bio-Inspired Computation*, **2(2)**, 78–84, (2010).
4. Y. Changjun, K.L. Teo, Y. Bai, *Optimization Letters*, **7(1)**, 23–38 (2013).
5. S. Lucidi, F. Rinaldi, *Optimization Letters*, **7(2)**, 297–307 (2013).
6. W. Murray, K.-M. Ng, *Computational Optimization and Applications*, **47(2)**, 257–288 (2010).
7. R.A. Shandiz, N. Mahdavi-Amiri, *American Journal of Operations Research*, **1(3)**, 185–189 (2011).