Proceedings of GOW'16, pp. 97 – 100.

Solving MINLP Problems by a Penalty Framework*

Ana Maria A.C. Rocha[1,2], M. Fernanda P. Costa[3] and Edite M.G.P. Fernandes[2]

[1]*Production and Systems Department, University of Minho, Braga, Portugal,* arocha@dps.uminho.pt

[2]*Algoritmi Research Centre, University of Minho, Braga, Portugal*

[3]*Centre of Mathematics, University of Minho, Braga, Portugal,* mfc@math.uminho.pt

Abstract A penalty framework for globally solving mixed-integer nonlinear programming problems is presented. Both integrality constraints and nonlinear constraints are handled separately by hyperbolic tangent penalty functions. The preliminary numerical experiments show that the proposed penalty approach is effective and the hyperbolic tangent penalties compete with other popular penalties.

Keywords: MINLP, Exact penalty, DIRECT

1. Introduction

A penalty approach for globally solving mixed-integer nonlinear programming (MINLP) problems is presented. A continuous relaxation of the MINLP problem is carried out by converting it to a finite sequence of bound constrained nonlinear programming (BCNLP) problems with only continuous variables. The MINLP problem is addressed in the form:

$$\begin{array}{ll} \min\limits_{x \in X \subset R^n} & f(x) \\ \text{subject to} & g_j(x) \leq 0, j = 1, \ldots, p \\ & h_l(x) = 0, l = 1, \ldots, m \\ & x_i \in \mathbb{R} \text{ for } i \in I_c \subseteq I \equiv \{1, \ldots, n\} \\ & x_j \in \mathbb{Z} \text{ for } j \in I_d \subseteq I \end{array} \tag{1}$$

where $f, g_j, h_l : \mathbb{R}^n \to \mathbb{R}$ are continuous possibly nonlinear functions in a compact subset of $\mathbb{R}^n$, herein defined as $X = \{x : -\infty < lb_i \leq x_i \leq ub_i < \infty, i = 1, \ldots, n\}$ and $I_c \cap I_d = \emptyset$ and $I_c \cup I_d = I$. Let C be the following subset of $\mathbb{R}^n$, $C = \{x \in X : g_j(x) \leq 0, j = 1, \ldots, p,\ h_l(x) = 0, l = 1, \ldots, m\}$ (that we assume to be compact) and let $W \subseteq C$ be the nonempty feasible region of the problem (1) $W = \{x \in C \subset \mathbb{R}^n : x_j \in \mathbb{Z} \text{ for } j \in I_d \subseteq I\}$. A penalty continuous formulation of the MINLP problem is used. First, a continuous relaxation of the MINLP problem (1) is obtained by relaxing the integrality conditions from $x_j \in \mathbb{Z}$, $j \in I_d$ to $x_j \in \mathbb{R}$, $j \in I_d$, and by adding a penalty term to the objective function that aims to penalize integrality constraint violation (see [2, 5]). Second, the resulting nonlinear programming (NLP) penalty problem is formulated as a BCNLP problem with an objective penalty function that is related to the objective function of the continuous relaxation of the MINLP and the nonlinear constraints violation.

Thus, our contribution in this article is directed to the combination of two penalty terms aiming to penalize integrality violation and nonlinear inequality and equality constraints violation separately. The penalty term for the integrality constraints is based on the hyperbolic tangent function [2] and the inequality and equality constraints violation is dealt with penalties that also rely on the hyperbolic tangent function. The solution of the BCNLP penalty prob-

*This work has been supported by COMPETE: POCI-01-0145-FEDER-007043 and FCT - Fundação para a Ciência e Tecnologia, within the projects UID/CEC/00319/2013 and UID/MAT/00013/2013.

lem are then obtained using the DIRECT algorithm [4], a deterministic algorithm for finding global solutions inside hyperrectangles. We illustrate the performance of the proposed exact penalty approach on three well-known test problems.

2. Penalty functions for MINLP

A penalty approach that can be extended to solve MINLP problems is investigated. In this context, a penalty function selected from a class of penalty functions for solving general integer problems [2, 5, 6] is used. Problem (1) is equivalent to the following continuous reformulation (in the sense that they have the same global minimizers), which comes out by relaxing the integer constraints on the variables and adding a particular penalty term to the objective function, as follows:

$$\begin{array}{ll} \min\limits_{x\in C} & \phi(x;\varepsilon) \equiv f(x) + P(x;\varepsilon) \\ \text{subject to} & x_i \in \mathbb{R},\, i = 1,\ldots,n, \end{array} \tag{2}$$

where $\varepsilon \in \mathbb{R}^+$ is a penalty parameter, and

$$P(x;\varepsilon) = \frac{1}{\varepsilon}\sum_{j\in I_d} \min_{lb_j \leq d_i \leq ub_j \,\wedge\, d_i \in \mathbb{Z}} \tanh\left(|x_j - d_i| + \varepsilon\right) \tag{3}$$

is the penalty term based on the hyperbolic tangent function, which is differentiable and strictly increasing on the set X [2]. The resulting penalty function in the NLP problem (2) is termed exact since $\exists \bar{\varepsilon} \in \mathbb{R}^+$ such that for all $\varepsilon \in (0, \bar{\varepsilon}]$, problems (1) and (2) have the same global minimizers (see Theorem 2.1 in [5]). Assuming that the set C is compact, the proof of Theorem 2.1 in [5] is based on specific assumptions on the objective function f and on the penalty term $P(x;\varepsilon)$. The particular case in (3) satisfies those assumptions (see Property 2.5 in [2]).

Furthermore, combining this idea with a penalty-based strategy for the nonlinear inequality and equality constraints, the BCNLP problem arises in the form

$$\begin{array}{ll} \min\limits_{x\in X} & \Psi(x;\varepsilon,\mu) \equiv \phi(x;\varepsilon) + \mu\left(\displaystyle\sum_{j=1}^{p} \tanh(\max\{g_j(x),0\}) + \sum_{l=1}^{m} \tanh(|h_l(x)|)\right) \\ \text{subject to} & x_i \in \mathbb{R},\, i = 1,\ldots,n, \end{array} \tag{4}$$

where we have extended the use of the '$\tanh(\cdot)$' to the general constraints violation and $\mu > 0$ is the penalty parameter. Ψ is a non-differentiable penalty function, although continuously differentiable at infeasible points, if f and the constraint functions are differentiable. An issue that remains to be established is the exactness property of the penalty function $\Psi(x;\varepsilon,\mu)$ in the context of using problem (4) to find an optimal solution to (2).

Algorithm 1 describes the proposed penalty framework aiming to find a global minimizer of the MINLP problem (1) by computing a global minimizer of the BCNLP problem formulated in (4), where $z^k \in X$, $z_j^k \in \mathbb{Z}, j \in I_d$ results from rounding x_j^k to the nearest integer and $z_i^k = x_i^k, i \in I_c$.

Besides forcing the integer variables to take integer values, another important issue is to reduce the overall nonlinear constraint violation, which is measured in terms of the maximum violation by $\eta(x^k) = \max_{j=1,\ldots,p;l=1,\ldots,m}\left\{\max\{g_j(x^k),0\}, |h_l(x^k)|\right\}$. Although more complex rules may be selected to control the reduction of parameters like $\varepsilon, \eta, \delta$ and the growth of parameter μ, we use simple schemes for these preliminary experiments.

To solve the BCNLP problems formulated in (4), a deterministic algorithm that uses only function evaluations, DIRECT [4] is used. DIRECT is efficient, in the sense that a few function evaluations are required, to find just an approximation to the solution, although the number of evaluations grows faster when a high quality solution is required. The problem to

Input: (x^*, f^*) (global solution), x^0, ε^1, μ^1, η^1, δ^1
Set $k = 1$;
while $\|x^{k-1} - x^*\| > 1E-3$ *or* $\eta(x^{k-1}) > 1E-4$ *or* $f^{k-1} > f^* + 1E-3$ **do**
  Compute x^k such that $\Psi(x^k;\varepsilon^k,\mu^k) \leq \Psi(x;\varepsilon^k,\mu^k) + \delta^k$, for all $x \in X$;
  if $\|x^k - z^k\| > 1E-3$ *and* $\phi(x^k;\varepsilon^k) - \phi(z^k;\varepsilon^k) \leq \varepsilon^k\|x^k - z^k\|$ **then**
    $\varepsilon^{k+1} = 0.1\varepsilon^k$; $\eta^{k+1} = \eta^k$; $\delta^{k+1} = \delta^k$;
  else
    if $\eta(x^k) \leq \eta^k$ **then**
      $\mu^{k+1} = \mu^k$; $\eta^{k+1} = \max\{0.1\eta^k, 1E-4\}$; $\delta^{k+1} = 0.1\delta^k$;
    else
      $\mu^{k+1} = 2\mu^k$; $\eta^{k+1} = \eta^k$; $\delta^{k+1} = \delta^k$;
    if $\|x^k - x^*\| > 1E-1$ **then**
      $\varepsilon^{k+1} = 0.9\varepsilon^k$;
  Set $k = k+1$;

Algorithm 1: Penalty-based algorithm

be addressed by DIRECT has the following form: for fixed $\varepsilon^k, \mu^k, \delta^k$, find $x^k \in X$ such that $\Psi(x^k;\varepsilon^k,\mu^k) \leq \Psi(x;\varepsilon^k,\mu^k) + \delta^k$ for all $x \in X$, assuming that the objective function $\Psi(x;\cdot)$ is Lipschitz continuous on X.

DIRECT is designed to completely explore the search space and is mainly characterized by sequentially dividing the space X into hyperrectangles and evaluating Ψ at their centers. To perform a balance between global and local search, the algorithm makes use of two important concepts: potentially optimal hyperrectangle and grouping according to size. The center c_i, the objective function value, $\Psi(c_i;\cdot)$, and the size d_i - originally given by the distance from the center to a corner - of the hyperrectangle i are used to define the groups of hyperrectangles, to select the potentially optimal hyperrectangles and divide them into smaller ones, until typically a maximum number of function evaluations is reached.

3. Numerical results

To make a preliminary evaluation of the practical behavior of the proposed penalty framework, based on the penalty presented in (4), we use three well-known MINLP problems (see [7]) which have two solutions, one global and one local:

$$(P1)\ \min\ f(x) \equiv -x_1 - x_2 \quad \text{s.t.}\ x_1x_2 - 4 \leq 0,\ 0 \leq x_1 \leq 4,\ x_2 \in \{0,\dots,6\},\quad f^* = -6.6666667$$

$$(P2)\ \min\ f(x) \equiv 35x_1^{0.6} + 35x_2^{0.6} \quad \text{s.t.}\ 600x_1 - 50x_3 - x_1x_3 + 5000 = 0,\ 600x_2 + 50x_3 - 15000 = 0,\ 0 \leq x_1 \leq 34, 0 \leq x_2 \leq 17,\ x_3 \in \{100,\dots,300\},\quad f^* = 189.311627$$

$$(P3)\ \min\ f(x) \equiv 2x_1 + x_2 \quad \text{s.t.}\ 1.25 - x_1^2 - x_2 \leq 0,\ x_1 + x_2 - 1.6 \leq 0,\ 0 \leq x_1 \leq 1.6,\ x_2 \in \{0,1\},\quad f^* = 2$$

In the context of the proposed penalty algorithm, we have also tested the three most popular general constraint penalties yielding the final penalty function:

$$\Psi(x;\varepsilon,\mu) = \phi(x;\varepsilon) + \mu\left(\sum_{j=1}^{p}(\max\{g_j(x),0\})^q + \sum_{l=1}^{m}(|h_l(x)|)^q\right) \quad \text{for } q = 1/2, 1, 2. \tag{5}$$

The penalty algorithm is coded in MATLAB programming language (Matlab Version 8.1.0.604 (R2013a)), the MATLAB code 'DIRECT.m' [3] is invoked, and the numerical experiments were carried out on a PC Intel Core 2 Duo Processor E7500 with 2.9GHz and 4Gb of memory.

Table 1 contains the results obtained by the present study with the penalty presented in (4) and with the penalty functions in (5) for comparison, where f is the computed solution, 'C.viol.' and 'I.viol.' are the general constraint and the integrality violations, respectively, Nf_{eval} is the number of function evaluations, It is the number of iterations and T is the CPU time (in seconds). For comparison, the results of a hybrid stochastic algorithm [1] and of an exact branch-and-reduce algorithm [7] are also shown. The herein listed results inside parentheses mean that the condition of the stopping rule of the algorithm related to that quantity is not satisfied. Our penalty algorithm always converges to the global solution and is able to reach good approximate solutions in a reasonable time. The practical performance of the penalty presented in (4) is comparable to the penalty in (5) with $q = 1$ and these two are superior to the other two penalties in comparison. It can be concluded that the proposed penalty approach for MINLP is effective and deserves further developments.

Table 1. Numerical results based on $\varepsilon^1 = 1$, $\mu^1 = 100$, $\eta^1 = 0.1$, $\delta^1 = 1$ and a maximum of 18 iterations.

Problem	Method	f	C.viol.	I.viol.	Nf_{eval}	It	T
$(P1)$	this study with (4)	-6.666661E+00	0.00E+00	5.65E-06	17643	1	3.9E+00
	penalty (5) and $q = 1/2$	-6.666661E+00	0.00E+00	5.65E-06	17717	1	4.0E+00
	penalty (5) and $q = 1$	-6.666661E+00	0.00E+00	5.65E-06	17643	1	3.7E+00
	penalty (5) and $q = 2$	-6.666661E+00	0.00E+00	5.65E-06	147756	8	3.1E+01
	in [1][a]	-6.666657E+00	0.00E+00	–	11513		3.3E+01
	in [7][b]	-6.666667E+00	–	–	–	–	7.0E-01[c]
$(P2)$	this study with (4)	(1.893756E+02)	3.53E-05	9.31E-04	170026	18	7.3E+01
	penalty (5) and $q = 1/2$	(2.016560E+02)	3.82E-07	(2.04E+00)	116382	18	5.5E+01
	penalty (5) and $q = 1$	(1.893756E+02)	3.66E-05	9.31E-04	175662	18	7.5E+01
	penalty (5) and $q = 2$	(1.893240E+02)	(1.41E-04)	(3.52E-03)	250082	18	1.0E+02
	in [1][a]	1.892946E+02	0.00E+00	–	13109		1.1E+02
	in [7][b]	1.893116E+02	–	–	–	–	7.0E-01[c]
$(P3)$	this study with (4)	2.000417E+00	0.00E+00	4.16E-04	13901	1	3.0E+00
	penalty (5) and $q = 1/2$	(2.027163E+00)	0.00E+00	(1.36E-02)	351368	18	7.3E+01
	penalty (5) and $q = 1$	2.000417E+00	0.00E+00	4.16E-04	13901	1	3.0E+00
	penalty (5) and $q = 2$	2.000395E+00	2.13E-05	4.37E-04	177651	11	3.7E+01
	in [1][a]	2.000000E+00	0.00E+00	–	4199		3.6E+01
	in [7][b]	2.000000E+00	–	–	–	–	7.0E-01[c]

[a] A multistart based Hooke-and-Jeeves filter method (best solution). [b] A branch-and-bound algorithm that relies on a domain reduction methodology. [c] CPU time in seconds on a Sun SPARC station 2.

References

[1] M.F.P. Costa, F.P. Fernandes, E.M.G.P. Fernandes, and A.M.A.C. Rocha. Multiple solutions of mixed variable optimization by multistart Hooke and Jeeves filter method. *Applied Mathematical Sciences*, 8(44):2163–2179, 2014.

[2] M.F.P. Costa, A.M.A.C. Rocha, R.B. Francisco, and E.M.G.P. Fernandes. Firefly penalty-based algorithm for bound constrained mixed-integer nonlinear programming. *Optimization*, 65(5):1085–1104, 2016.

[3] D.E. Finkel. DIRECT Optimization Algorithm User Guide. Report, Center for Research in Scientific Computation, North Carolina State University, 2003.

[4] D.R. Jones, C.D. Perttunen, and B.E. Stuckmann. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.

[5] S. Lucidi and F. Rinaldi. Exact penalty functions for nonlinear integer programming problems. *Journal of Optimization Theory and Applications*, 145(3):479–488, 2010.

[6] S. Lucidi and F. Rinaldi. An exact penalty global optimization approach for mixed-integer programming problems. *Optimization Letters*, 7(2):297–307, 2013.

[7] H.S. Ryoo and N.V. Sahinidis. Global optimization of nonconvex NLPs and MINLPs with applications in process design. *Computers and Chemical Engineering*, 19(5):551–566, 1995.