Numerical Experiments with Nonconvex MINLP Problems

Florbela P. Fernandes*,†, M. Fernanda P. Costa** and Edite M.G.P. Fernandes‡

*Department of Mathematics, Polytechnic Institute of Braganca, Portugal
†CMAT-University of Minho, Portugal
**Department of Mathematics and Applications, University of Minho, Portugal
‡Department of Production and Systems, University of Minho, Portugal

Abstract. We present a methodology to solve nonconvex Mixed-Integer Nonlinear Programming problems, that combines the Branch-and-Bound and simulated annealing type methods, which was implemented in MATLAB. A set of benchmark functions with simple bounds and different dimensions was used to analyze its practical behaviour. We exhibit computational results showing the good performance of the method.

Keywords: Mixed-Integer Programming, Branch-and-Bound, Stochastic Method.
PACS: 02.60.Pn

INTRODUCTION

We aim to develop a methodology that is able to efficiently solve nonlinear optimization problems that involve discrete as well as continuous variables, usually denoted by mixed-integer nonlinear programming (MINLP) problems. This particular type of problems can be represented by the following form:

$$\begin{aligned}
\min \quad & f(x,y) \\
\text{s. t.} \quad & g_j(x,y) \leq 0, \ j \in J \\
& x \in X, y \in Y
\end{aligned} \tag{1}$$

where $X \subseteq \mathbb{R}^n$ is assumed to be a convex compact set, $Y \subseteq \mathbb{Z}^p$ corresponds to a polyhedral set of integer points, and the nonlinear functions $f : \mathbb{R}^{n+p} \to \mathbb{R}$ and $g : \mathbb{R}^{n+p} \to \mathbb{R}^m$ are assumed to be continuously differentiable. The set J is the index set of inequality constraints, and x and y are the continuous and discrete/integer variables, respectively. If the objective function f and the constraint functions g are convex, the problem is known as convex, otherwise the problem is a nonconvex MINLP [3]. A great variety of problems in the areas of engineering design, computational chemistry, computational biology, communications and finance can be represented by MINLP models. In particular, one may find applications which include gas network problems, nuclear core reloaded problems, cyclic scheduling trim-loss optimization in the paper industry, synthesis problems, layout problems and thermal insulation systems [1, 2]. Due to the frequent appearance of MINLP in industrial processes, it is crucial to develop solution techniques with reduced computational requirements. Until now, significant progress has been made in the solution techniques for convex MINLP [1, 4, 11]. However, techniques and solvers for nonconvex MINLP problem have just started to appear in the literature [9, 10].

Our proposal combines two strategies: a Branch-and-Bound (B&B) method to search for integer solutions and a simulated annealing type method to promote convergence to global solutions of the nonconvex nonlinear programming (NLP) relaxation problems that appear in the nodes of the B&B tree search.

THE METHOD IN STUDY

Our preliminary studies consider bound constrained nonconvex MINLP problems with the form:

$$\min_{x \in X, y \in Y} f(x,y) \tag{2}$$

where $X = \{x \in \mathbb{R}^n : l_x \leq x \leq u_x\}$ with $l_x, u_x \in \mathbb{R}^n$ and $Y = \{y \in \mathbb{Z}^p : l_y \leq y \leq u_y\}$ with $l_y, u_y \in \mathbb{Z}^p$.

The herein presented methodology to solve problem in Eq. (2) uses a B&B scheme and a simulated annealing algorithm to guarantee convergence, at least with probability one, to a global optimum of the nonconvex NLP relaxation problem (that arises in each node of a tree in the B&B algorithm). A brief description of the two strategies combined in the herein proposed method is presented below.

Branch-and-Bound method

The B&B methodology can be explained in terms of a tree search [7, 8]. Initially, all integer variables are relaxed and the resulting NLP relaxation problem is solved. If all the integer variables of the problem take integer values, the optimal solution is found and the algorithm stops. Otherwise, a tree search is performed in the space of the integer variables. Simple bounds are added successively (at nodes of the tree) yielding subproblems at each kth step in the branch and bound search. The solution of each subproblem provides a lower bound for the subproblems in the descent nodes of the tree. This process continues until the lower bound exceeds the current upper bound, the NLP subproblem is infeasible, or the solution provides integer values for the integer variables. The integer solutions (at the nodes of the tree) give upper bounds on the optimal integer solution. This process continues until there are no more nodes to explore.

The SAHPS method

To solve the nonconvex NLP relaxation problem at each node of the B&B tree search a simulated annealing heuristic pattern search (SAHPS) [6] is used. The simulated annealing (SA) approach is combined with the heuristic pattern search to form the hybrid method SAHPS [6]. The SAHPS tries to get better movements through the SA acceptance procedure or by using a heuristic pattern search (HPS) procedure. More specifically, a new exploring neighborhood search is introduced to generate a number of SA trial points. If some of these trial points can be accepted by the SA acceptance procedure, this means that the search can go further and there is no need to use a local search method. Otherwise, some iterations of the HPS method are applied to generate more local exploratory trial points. The HPS method is based on two main ideas. First, it uses a derivative-free heuristic method to produce an approximate descent direction at the current solution – the therein denoted approximate descent direction (ADD) method. Next, uses the ADD method to design a new pattern search method called the HPS. The ADD method is recalled to obtain an approximate descent direction v at the current iterate x_k. The direction v is computed at x_k, after generating m points $\{y_i\}_{i=1}^m$ close to x_k, as:

$$v = \sum_{i=1}^{m} w_i e_i \tag{3}$$

where

$$w_i = \frac{\Delta f_i}{\sum_{j=1}^{m} |\Delta f_j|}, \quad \Delta f_i = f(y_i) - f(x_k), \quad i = 1, \ldots, m, \quad e_i = -\frac{y_i - x_k}{\|y_i - x_k\|} \quad i = 1, \ldots, m.$$

If no improvement is obtained along the vector v, then this is used to prune a set D of pattern search directions to generate other exploratory moves. In the final stage of the search, a direct search method is applied to refine the best solution obtained so far. The algorithm uses a modified version of the Nelder-Mead method. All the details about this method can be found in [6].

NUMERICAL RESULTS

The proposed method, herein called BBSAHPS, was implemented in MATLAB™ and uses a B&B method combined with the SAHPS method of Hedar and Fukushima [6]. The MATLAB code of SAHPS was integrated in our solver and called inside our B&B algorithm (to solve the relaxed NLP in each node of the tree). The values of the parameters inside the SAHPS are the default values as described in [6]. Some minor changes were made to incorporate the SAHPS method into the B&B algorithm. A collection of six test functions was used to analyze the practical behavior

of BBSAHPS [5, 6]. Further, since the mathematical formulation of the chosen problems vary with the dimension n, we also aim to study the effect of the number of variables on the performance of the proposed method. The tested values are $n = 2, n = 4$ and $n = 10$ giving a total of 18 instances. The set of test functions, with the feasible region and the global minimum are displayed in Table 1.

TABLE 1. Test functions

Ackley function (A)	Dixon and Price function (DP)
$A(x) = -20e^{-0.2\sqrt{1/n\sum_{i=1}^{n} x_i^2}} - e^{1/n\sum_{i=1}^{n}\cos(\pi x_i)} + 20 + e$	$DP(x) = (x_1 - 1)^2 + \sum_{i=2}^{n} i(2x_i^2 - x_{i-1})^2$
$x \in [-15, 30]^n$ and $A(x^*) = 0$	$x \in [-10, 10]^n$ and $DP(x^*) = 0$

Rastrigin function (R)	Sphere function (S)
$R(x) = 10n + \sum_{i=1}^{n} x_i^2 - 10\cos(2\pi x_i)$	$S(x) = \sum_{i=1}^{n} x_i^2$
$x \in [-3, 6]^n$ and $R(x^*) = 0$	$x \in [-3, 6]^n$ and $S(x^*) = 0$

Trid function (T)	Levy function (L)
$T(x) = \sum_{i=1}^{n} (x_i - 1)^2 - \sum_{i=2}^{n} x_i x_{i-1}$	$L(x) = \sin^2(\pi y_1) + \sum_{i=1}^{n-1}\left((y_i - 1)^2\left(1 + 10\sin^2(\pi y_{i+1})\right)\right) +$
	$+ (y_n - 1)^2\left(1 + \sin^2(2\pi y_n)\right)$
$x \in [-n^2, n^2]^n$	where $y_i = 1 + \dfrac{x_i - 1}{4}$, for $i = 1, \cdots, n$
$T_2(x^*) = -2, T_4(x^*) = -16, T_{10}(x^*) = -210$	$x \in [-10, 10]^n$ and $L(x^*) = 0$

For each function the solver was run 100 times, using the same initial point x_0. Table 2 summarizes the BBSAHPS results obtained for each test problem. In this table, the best of the objective function values obtained after the 100 runs, "Best f", is reported for each test problem. In order to show more details concerning the quality of the obtained solutions, the average of the best function values, "Average f", and the standard deviation of the best obtained function values, "σ", are also reported in Table 2. Further, the number of variables, "n", the success rate of obtaining the global minimum, "% success", the average CPU time over the 100 runs, "time (s)" (in seconds), the average number of major cycles of the B&B tree, "cycles", and the average number of function evaluations, "f eval.", are shown in columns 2 – 6 of Table 2.

TABLE 2. Numerical results obtained during the 100 runs and using BBSAHPS

f	n	% success	Average time (s)	cycles	f eval.	Best f	Average f	σ
A	2	100	0.09	8	802	8.88178×10^{-16}	1.6826×10^{-07}	1.68265×10^{-07}
	4	88	0.39	16	1281	8.88178×10^{-16}	1.75721×10^{-07}	4.68973×10^{-07}
	10	43	3.54	41	3136	8.88178×10^{-16}	5.28589×10^{-08}	2.07519×10^{-07}
DP	2	62						
		36	0.08	6	744	8.08397×10^{-16}	4.17957×10^{-9}	6.60461×10^{-9}
		26	0.08	6	733	8.78736×10^{-14}	3.61570×10^{-9}	3.66593×10^{-9}
	4	90						
		87	0.25	7	1626	1.56997×10^{-09}	1.63582×10^{-09}	3.57117×10^{-10}
		3	0.20	7	1615	6.66341×10^{-11}	4.03201×10^{-09}	3.7783×10^{-09}
L	2	95	0.11	11	964	1.49976×10^{-32}	1.49976×10^{-32}	1.92600×10^{-47}
	4	86	0.60	24	2507	1.49976×10^{-32}	2.04079×10^{-15}	1.89255×10^{-14}
	10	64	10.75	61	11681	1.49976×10^{-18}	1.52759×10^{-13}	1.52759×10^{-13}
R	2	100	0.03	2	333	$0.00000 \times 10^{+00}$	3.11857×10^{-13}	2.91543×10^{-12}
	4	99	0.59	17	1261	$0.00000 \times 10^{+00}$	1.08864×10^{-12}	1.08318×10^{-11}
	10	100	5.3	41	3137	$0.00000 \times 10^{+00}$	5.79092×10^{-15}	5.79092×10^{-14}
S	2	96	0.07	9	683	$0.00000 \times 10^{+00}$	9.25469×10^{-15}	5.14343×10^{-14}
	4	84	0.39	17	1257	$0.00000 \times 10^{+00}$	1.47969×10^{-14}	1.04014×10^{-13}
	10	55	6.28	46	5747	$0.00000 \times 10^{+00}$	5.73489×10^{-14}	1.68233×10^{-13}
T	2	91	0.12	12	1468	$-2.00000 \times 10^{+00}$	$-2.00000 \times 10^{+00}$	1.16208×10^{-13}
	4	67	0.62	24	2098	$-1.60000 \times 10^{+01}$	$-1.60000 \times 10^{+01}$	2.03429×10^{-13}
	10	79	18.00	80	14857	$-2.10000 \times 10^{+02}$	$-2.10000 \times 10^{+02}$	4.84208×10^{-13}

Some runs (out of the 100) converged to different global minimizers. For example, when solving Dixon and Price (DP) function with $n = 4$, 87 runs (out of 100) converged to one global minimizer and in 3 runs (out of 100) the solver converged to the other global minimizer. The overall percentage of successful runs – converging to a global solution with mixed-integer variable – is then 90. When using $n = 10$ with this test function, the method did not converge to any of the global minimizers. The results obtained by the BBSAHPS method are quite satisfactory. With all these functions the BBSAHPS method could find the global minimum, taking into account the integrality of some variables. The accuracy of the achieved solutions, in terms of "Best f", is very good and the consistency of our proposed stochastic B&B-type method is high, since the standard deviations of the f values are close to zero. As shown in Table 2, the computational cost in terms of CPU time required by the proposed BBSAHPS is rather small for problems with $n = 2$ or $n = 4$. Problems with 10 variables are more CPU demanding. We observe that the increase in CPU time and number of function evaluations is not proportional to the number of variables. In general, we may say that the BBSAHPS method behaves quite well with the six tested functions (with variable dimensions) and it has a success rate superior to 79% for at least 12 instances, as shown in Table 2.

CONCLUSIONS AND FUTURE WORK

In this paper we have presented the BBSAHPS method in which a Branch-and-Bound procedure is combined with a hybrid global search to find the minimum of nonconvex MINLP problems with simple bounds. BBSAHPS method was implemented using MATLAB™ and some results are shown considering six test functions. This method is able to find the global optimum, with integer restrictions for some variables in all but one tested instances, with different dimensions. The performance of the BBSAHPS method is quite satisfactory. Future developments will be focused on an efficient constraint-handling technique that uses the concept of dominance present in multi-objective optimization, known as filter method. This type of methodology is now being incorporated in the SAHPS method to be able to solve problems of type (1).

ACKNOWLEDGMENTS

REFERENCES

1. K. Abhishek, S. Leyffer and J. Linderoth, FilMINT: An Outer-Approximation-Based Solver for Nonlinear Mixed Integer Programs. Technical Report ANL/MCS-P1374-0906, Mathematics and Computer Science Division, Argonne National Laboratory, 2006.
2. K. Abhishek, S. Leyffer and J.T. Linderoth, Modeling without categorical variables: a mixed-integer nonlinear program for the optimization of thermal insulation systems, *Optimization and Engineering*, 11, 185–212, 2010.
3. L. Biegler and I. Grossmann, Retrospective on optimization, *Computers and Chemical Engineering*, 28, 1169-1192, 2004.
4. P. Bonami, L. Biegler, A. Conn, G. Cornuejols, I. Grossmann, C. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya and A. Wachter, An algorihmic framework for convex mixed integer nonlinear programs, Technical Report RC23771, IBM Research Report, 2005.
5. M.J. Hirsch, P.M. Pardalos, M.G.C. Resende, Speeding up continuous GRASP, *European Journal of Operational Research*, 205, 507-521, 2010.
6. A. Hedar and M. Fukushima, Heuristic pattern search and its hybridization with simulated annealing for nonlinear global optimization, *Optimization Methods and Software*, 19, 291-308, 2004.
7. S. Leyffer, Deterministic Methods for Mixed Integer Nonlinear Programming. PhD Thesis, University of Dundee, United Kingdom, 1993.
8. S. Leyffer, Integrating SQP and branch and bound for mixed integer nonlinear programming. *Computational Optimization and Applications*, 18, 295-309, 2001.
9. S. Leyffer, A. Sartenaer, and E. Wanufelle, Branch-and-Refine for Mixed Integer Nonconvex Global Optimization, Preprint ANL/MCS-P1547-0908, Mathematics and Computer Science Division, Argonne National Laboratory, 2008.
10. I. Nowak and S Vigerske, LaGO: a (heuristic) branch and cut algorithm for nonconvex MINLPs, *Central European Journal of Operations Research*, 16, 127-138, 2008.
11. M.H. Van der Vlerk, Convex approximations for a class of mixed-integer recourse models, *Annals of Operations Research*, 177, 139–150, 2010.