

Overview on Mixed Integer Nonlinear Programming 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*

^{**}*Department of Mathematics for Science and Technology, University of Minho, Portugal*

[‡]*Department of Production and Systems, University of Minho, Portugal*

Abstract. Many optimization problems involve integer and continuous variables that can be modeled as mixed integer nonlinear programming (MINLP) problems. This has led to a wide range of applications, in particular in some engineering areas. Here, we provide a brief overview on MINLP, and present a simple idea for a future nonconvex MINLP solution technique.

Keywords: mixed-integer nonlinear programs; branch-and-bound; outer-approximation.

PACS: 02.60.Pn

INTRODUCTION

Many optimization problems involve discrete and continuous variables that can be modeled as mixed integer nonlinear programming (MINLP) problems. For instance, integer variables can represent the number of workers needed to perform a certain task whereas continuous variables can denote physical values, such as pressure or temperature. This has led to a wide range of applications in the field of process systems engineering [10]. 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 [1]. Other examples are efficient management of electricity transmission, contingency analysis and blackout prevention of electric power systems [15].

A mixed integer nonlinear program formulation can be represented as:

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

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, $f: \mathbb{R}^{n+p} \rightarrow \mathbb{R}$ and $g: \mathbb{R}^{n+p} \rightarrow \mathbb{R}^m$ are continuously differentiable functions, 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. In the last decades, significant progress has been made in the solution techniques for convex MINLP. However, techniques and solvers for nonconvex MINLP problem have just started to appear in the literature [14].

The paper is organized as follows. First, we briefly present the main techniques for solving convex MINLP problems. We then present a future idea for nonconvex MINLP based on a heuristic linear/quadratic interpolation model-based method.

MINLP SOLUTION TECHNIQUES

Here, we assume that the problem is a convex MINLP. The available methods to solve this type of problem can be derived from three basic nonlinear programming (NLP) subproblems and from one cutting plane mixed integer linear programming (MILP) problem, related with (1) [2, 10]. Different methodologies have been proposed to solve convex MINLP, in particular: Branch-and-Bound (BB), Outer-Approximation (OA), LP/NLP-based branch-and-bound

(LP/NLP-BB), Generalized Benders Decomposition, Extended Cutting Plane, and Branch-and-Cut, [9, 10, 11, 15, 19]. Our brief presentation below will focus mainly on the BB, OA and LP/NLP-BB methodologies.

Branch-and-Bound. Although BB was originally devised for MILP, it can also be applied to MINLP. We refer the reader to Leyffer [13] and the references therein included. The method solves a NLP relaxation subproblem at each node of the tree, instead of an LP, represented by

$$\begin{aligned} LB^k \equiv \min \quad & f(x, y) \\ \text{s. t.} \quad & g_j(x, y) \leq 0, \quad j \in J \\ & x \in X, y \in Y_R \\ & y_i \leq u_i^k, \\ & y_i \geq l_i^k \end{aligned} \tag{NLP-rel-k}$$

for some i , where Y_R is a continuous relaxation of the set Y and $u_i^k = \lfloor y_i^h \rfloor$ (floor function applied to the noninteger value y_i^h of a previous step $h < k$), $l_i^k = \lceil y_i^m \rceil$ (ceiling function applied to the noninteger value y_i^m of a previous step $m < k$). When $k = 0$, problem (NLP-rel-k) corresponds to the continuous NLP relaxation of the MINLP problem (1). The BB method starts by solving the subproblem (NLP-rel-k) with $k = 0$. 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 $y_i \leq u_i^k, y_i \geq l_i^k$ are added successively (at nodes of the tree) yielding subproblems (NLP-rel-k) at each k th step in the branch and bound search. The solution of each subproblem (NLP-rel-k) 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. An example of the enumeration tree is represented on the left of Figure 1. Double boxed nodes represent integer solutions. The optimal solution is LB^5 . The BB method is only attractive if the NLP relaxation subproblems are inexpensive to solve, or when only a few of them need to be solved. Otherwise the tree can grow very large and solving a large number of NLP can be time-consuming. The well-known solvers Bonmin [3], KNITRO [4], MINLP-BB [12] and SBB [8], are branch-and-bound based methodologies and available for convex MINLP.

Outer-Approximation. The OA method was introduced by Duran and Grossman [6] and later used by Fletcher and Leyffer [7]. The OA method performs a cycle of major iterations, $k = 1, \dots, K$, in which the NLP reduction subproblem (NLP-red-k), for a given y^k ,

$$\begin{aligned} f_U^k \equiv \min \quad & f(x, y^k) \\ \text{s. t.} \quad & g_j(x, y^k) \leq 0, \quad j \in J \\ & x \in X \end{aligned} \tag{NLP-red-k}$$

is solved, being x^k the solution, and the MILP relaxation master problem (M-OA):

$$\begin{aligned} LB^K \equiv \min \quad & l \\ \text{s. t.} \quad & f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \leq l \\ & g_j(x^k, y^k) + \nabla g_j(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \leq 0, \quad j \in J \\ & x \in X, y \in Y \\ & k = 1, \dots, K \end{aligned} \tag{M-OA}$$

is updated with the corresponding objective and constraint function linearizations at the point (x^k, y^k) , and solved to generate a new vector of discrete values. Usually, not all $y^k \in Y$ give rise to a feasible (NLP-red-k) subproblem, i.e., does not exist $x^k \in X$ that satisfies $g_j(x^k, y^k) \leq 0, j \in J$. If subproblem (NLP-red-k) is feasible, the solution x^k is used to define the first MILP master problem. Otherwise, the following feasibility NLP subproblem

$$\begin{aligned} \min \quad & u \\ \text{s. t.} \quad & g_j(x, y^k) \leq u, \quad i \in J \\ & x \in X, u \in \mathbb{R}^+ \end{aligned} \tag{NLP-F}$$

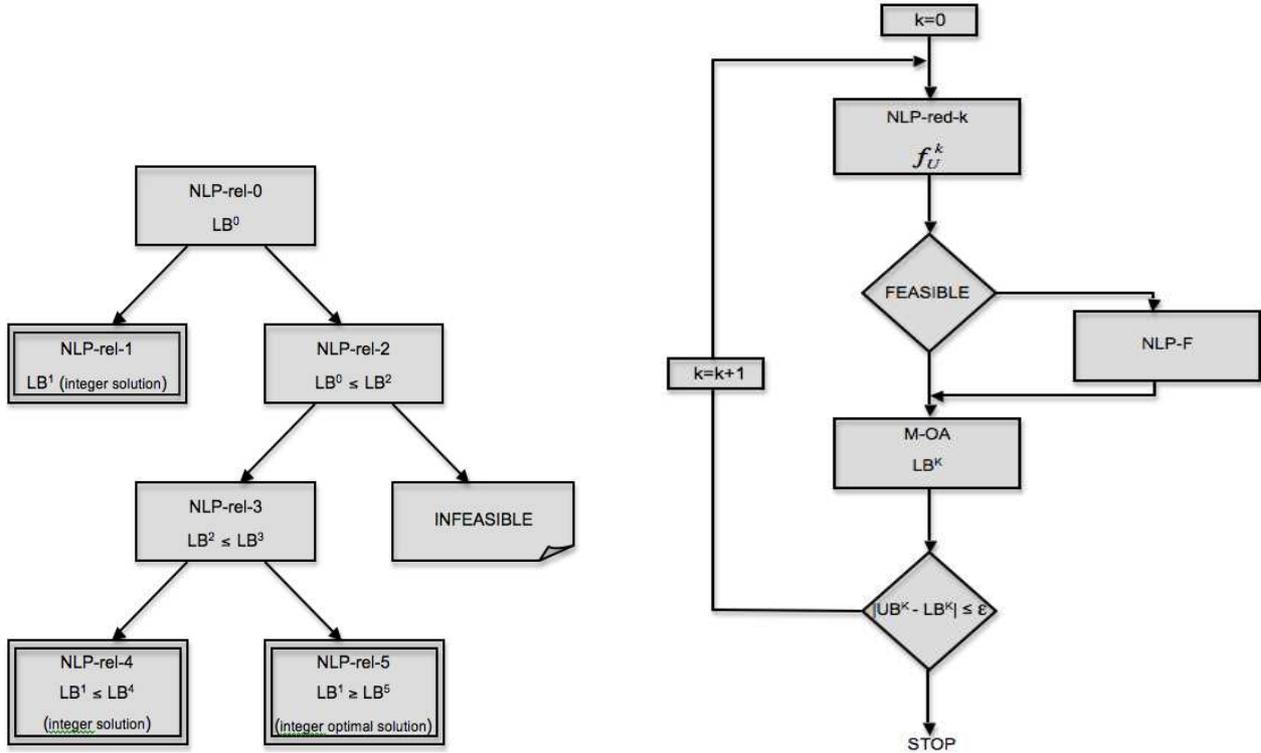


FIGURE 1. Enumeration tree in branch-and-bound (on the left) and flowchart of outer-approximation (on the right)

for a fixed y^k , is solved to provide the corresponding x^k . The feasibility NLP subproblem (NLP-F) can be interpreted as the minimization of the infinity-norm measure of infeasibility of the corresponding NLP subproblem. The OA algorithm starts with a vector (x^0, y^0) that can be either a feasible solution to (1) or to the continuous NLP relaxation of the MINLP problem. The solution of problem (M-OA) LB^k corresponds to a lower bound for problem (1). Since the linearizations are accumulated as k increases, the master problems (M-OA) generate a sequence of lower bounds $LB^1 \leq LB^2 \leq \dots \leq LB^k$. This cycle of iterations continues until the difference between the upper bound $UB^k = \min_k f_U^k$, provided by subproblems (NLP-red-k), and the lower bound provided by the MILP relaxation master problem LB^k is sufficiently small. The flowchart on the right of Figure 1 shows the major steps in OA type algorithms.

The successive accumulation of linearizations generating master problems (M-OA) with large number of constraints is the main drawback of OA based methods. Keeping only the last linearization point is not appropriate since the monotonic increase of the lower bound is not guaranteed. It is shown in [18] that the aggregation of linear approximations to the objective and constraint functions can be carried out for the efficient solving of MILP master problem. Experience shows that the OA method requires in general a reduced number of major iterations. If f and g are linear functions in x and y then OA method converges in one iteration, since the MILP master program (M-OA) corresponds to the problem (1). Furthermore, the master problem (M-OA) need not be solved to optimality. It is sufficient to find a new point (y^k, x^k) that gives a solution to the MILP that is below UB^k within a specified tolerance.

LP/NLP-based Branch-and-Bound. The LP/NLP-BB is a clever extension of OA, which avoids solving an alternating sequence of MILP master problems (M-OA) and NLP subproblems. The basic idea consists of performing an LP-based branch-and-bound for the continuous relaxation to (M-OA) and enforcing integrality of the variables y by branching. This approach avoids solving multiple MILP master problems by interrupting the MILP tree search whenever an integer feasible solution y^k , say, is found to solve the NLP subproblem ((NLP-red-k) or (NLP-F)). Then it updates the representation of the master problem in the current open nodes of the tree with the corresponding linearizations, and the MILP tree search continues. The algorithm is stated in full detail in [11]. A list of some available solvers based on linearizations is: Bonmin [3], Dicopt (<http://www.gams.com/dd/docs/solvers>) and FilMINT [1].

Recent and future work

Branch-and-bound type methods have already been proposed for solving nonconvex MINLP [14]. The list of the available software, mostly deterministic optimizers for nonconvex MINLP is the following: BARON (<http://archimedes.cheme.cmu.edu/baron/baron.html>), Couenne (<https://projects.coin-or.org/Couenne>) and LaGO [16]. Most of these codes are available for use free of charge and/or are open source. The difficulty with nonconvex MINLP is that common optimizers when solving NLP problems with nonconvex f and g functions are able to guarantee convergence only to a local minimum. This means that the solution of the NLP relaxation subproblem does not provide a lower bound on the solution of the original problem (1). An additional relaxation process is required, typically involving the decomposition of the nonlinear functions into components, see [15].

An alternative idea, keeping a classic branch-and-bound approach, is to solve nonconvex NLP subproblems using a heuristic combination of deterministic and stochastic algorithms, that is able to guarantee convergence to a global solution with a high probability. This is a matter for future research. Since our proposal aims to focus on derivative-free methods, the required linear and/or quadratic approximation models are typically constructed via interpolation model-based methods [5, 17].

ACKNOWLEDGMENTS

This work has been partially supported by the FCT (Fundação para a Ciência e a Tecnologia), Portugal.

REFERENCES

1. Abhishek, K., Leyffer, S. and Linderoth, J. (2006). 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.
2. Biegler, L. and Grossmann, I. (2004). Retrospective on optimization. *Computers and Chemical Engineering* 28: 1169–1192.
3. Bonami, P., Biegler, L., Conn, A., Cornuejols, G., Grossmann, I., Laird, C., Lee, J., Lodi, A., Margot, F., Sawaya, N., Wachter, A., (2005). An algorithmic framework for convex mixed integer nonlinear programs. Technical Report RC23771, IBM Research Report.
4. Byrd, R. H., Nocedal, J. and Waltz, R. A. (2006). KNITRO: An Integrated Package for Nonlinear Optimization in Large-Scale Nonlinear Optimization, G. di Pillo and M. Roma, eds, 35-59, Springer-Verlag.
5. Fasano, G., Morales, J. L., Nocedal, J. (2009) On the geometry phase in model-based algorithms for derivative-free optimization. *Optimization Methods and Software* 24: 145–154.
6. Duran, M. and Grossmann, I. (1986). An outer approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming* 36: 307–339.
7. Fletcher, F. and Leyffer, S. (1994). Solving mixed-integer nonlinear programs by outer approximation. *Mathematical Programming* 66: 327–349.
8. Gams Development Corp. (2003). Gams – The Solver Manuals. *GAMS Development Corp.*, Washington, DC.
9. Geoffrion, A. M. (1972). Generalized Benders decomposition. *Journal of Optimization Theory and Applications* 10:4 237–260.
10. Grossmann, I. (2002). Review of Nonlinear Mixed-Integer and Disjunctive Programming Techniques. *Optimization and Engineering* 3: 227–252.
11. Leyffer, S. (1993). Deterministic Methods for Mixed Integer Nonlinear Programming. PhD Thesis, University of Dundee, United Kingdom.
12. Leyffer, S. (1999). User Manual for MINLP BB, University of Dundee
13. Leyffer, S. (2001). Integrating SQP and branch and bound for mixed integer nonlinear programming. *Computational Optimization and Applications* 18: 295–309.
14. Leyffer, S., Sartenaer, A., Wanufelle, E. (2008). Branch-and-Refine for Mixed Integer Nonconvex Global Optimization. Preprint ANL/MCS-P1547-0908, Mathematics and Computer Science Division, Argonne National Laboratory.
15. Leyffer, S., Linderoth, J., Luedtke, J., Miller, A., Munson, T. (2009). Applications and Algorithms for Mixed Integer Nonlinear Programming. Preprint ANL/MCS-P1630-0509, Mathematics and Computer Science Division, Argonne National Laboratory.
16. Nowak, I., Vigerske, S. (2008). LaGO: a (heuristic) branch and cut algorithm for nonconvex MINLPs. *Central European Journal of Operations Research* 16: 127–138.
17. Powell, M. J. C. (2004). Least Frobenius norm updating of quadratic models that satisfy interpolation conditions. *Mathematical Programming* 100: 183–215.
18. Quesada, I., Grossmann, I. E. (1992). An LP/NLP based branch and bound algorithm for convex MINLP optimization problems. *Computers and Chemical Engineering* 16:2 937–947.
19. Westerlund, T., Pettersson, F. (1995). An Extended cutting plane method for solving convex MINLP problems. *Computers and Chemical Engineering* 19: 131–136.