Proceedings of GOW 2012, pp. 1 – 4.

A hybrid algorithm between branch-and-bound and outer approximation for Mixed Integer Nonlinear Programming

Wendel A. X. Melo[1], Marcia H. C. Fampa[2], and Fernanda M. P. Raupp[3]

[1] *Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil,* wendelmelo@cos.ufrj.br

[2] *Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil,* fampa@cos.ufrj.br

[3] *Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brazil,* fraupp@puc-rio.br

Abstract In this work, we present a new hybrid algorithm for convex Mixed Integer Nonlinear Programming combining branch-and-bound and outer approximation algorithms in an effective and efficient way.

Keywords: mixed integer nonlinear programming, branch-and-bound, outer approximation, hybrid algorithm

1. Introduction

Mixed Integer Nonlinear Programming (MINLP) problems are characterized by the presence of nonlinear functions of continuous and discrete variables. The MINLP problem addressed in this work can be algebraically represented in the following way:

$$
\begin{array}{lll}
(P) & \text{minimize}_{x,y} & f(x,y) \\
 & \text{s. t.:} & g(x,y) \ \leq \ 0 \\
 & & x \in X, \ y \in Y \cap \mathbb{Z}^{n_y}
\end{array}
\tag{1}
$$

where X and Y are polyhedral subsets of $\mathbb{R}^{n_x}$ and $\mathbb{R}^{n_y}$, respectively, and Y is bounded. The functions $f : X \times Y \to \mathbb{R}$ and $g : X \times Y \to \mathbb{R}^m$ are convex and twice continuously differentiable. We call problem (1) by P and its continuous relaxation by $\tilde{P}$.

Algorithms in two distinct methodological classes have been employed to solve P: outer approximation algorithms [2] and branch-and-bound algorithms (the reader interested in MINLP algorithms can see [3, 4]). In [1], a hybrid approach combining algorithms in these two classes was introduced. In this work, we propose a new hybrid algorithm combining also the two cited methodologies in a more effective way than [1]. Our main goal is to potentialize the particular advantages of each class and remediate theirs drawbacks. In Section 2, we present an outer approximation algorithm and in Section 3, we show the proposed hybrid approach.

2. Outer Approximation

Proposed by Duran and Grossmann in [2], the Outer Approximation (OA) algorithm alternates between solving a Mixed Integer Linear Programming problem (MILP) and one or two NonLinear Programming problems (NLP). Its main idea is to approximate P by the following MILP problem that is built using linearization of functions in P on a set T of t linearization points, i.e., $T = \{(x^0, y^0), (x^1, y^1), \ldots, (x^t, y^t)\}$:

$$
\begin{array}{llll}
\left(P^{OA}(T)\right) & \min_{\alpha,x,y} & \alpha & \\
& \text{s. t.:} & \nabla f(x^k,y^k)^T \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} + f(x^k,y^k) \ \leq \ \alpha, & \forall (x^k,y^k) \in T \\
& & \nabla g(x^k,y^k)^T \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} + g(x^k,y^k) \ \leq \ 0, & \forall (x^k,y^k) \in T \\
& & x \in X, \ y \in Y \cap \mathbb{Z}^{n_y}. &
\end{array}
\tag{2}
$$

As P is convex, we notice that problem (2) is a relaxation of P, which provides valid lower bounds to P. The baseline of OA algorithm is showed in Algorithm 1. New linearization points are added to set T, as the algorithm evolves. This strengthens the relaxation given by (2) and generates a non-decreasing sequence of lower bounds to P.

Let $(\hat{x}, \hat{y})$ be an optimal solution of an instance of problem (2). The integer variable values $\hat{y}$ are used to build the following NLP problem from P:

$$
\begin{array}{lll}
(P_{\hat{y}}) & \text{minimize}_x & f(x,\hat{y}) \\
& \text{s. t.:} & g(x,\hat{y}) \ \leq \ 0 \\
& & x \in X.
\end{array}
\tag{3}
$$

Suppose problem (3) is feasible and let $\bar{x}$ be an optimal solution. So, the point $(\bar{x}, \hat{y})$ provides an upper bound to P. Thus, OA algorithm adds this point to the set of linearization points T and starts a new iteration using as stopping rule the annulment of the optimality gap.

In the case problem (3) is infeasible, OA algorithm solves the following feasibility problem:

$$
\begin{array}{lll}
(P_{\hat{y}}^V) & \text{minimize}_{u,x} & \sum_{i=1}^m u_i \\
& \text{s. t.:} & g(x,\hat{y}) \ \leq \ u \\
& & x \in X, \ u \in (\mathbb{R}^+)^m
\end{array}
\tag{4}
$$

Let $(\breve{u}, \breve{x})$ be an optimal solution of (4) in the described context. Then, the point $(\breve{x}, \hat{y})$ is added to the set T. Conforming demonstrated in [2], if the KKT conditions are satisfied at the optimal solutions of (3) and (4), OA algorithm converges in a finite number of iterations.

ALGORITHM 1: Outer approximation ;
INPUT: P: Problem (1), T_0: initial set of linearization points (it can be empty) ;
OUTPUT: (x^*, y^*): optimal solution of P ;

```
z^U = +∞ ; z^L = −∞ ;
Let (x^0, y^0) be an optimal solution of P̃ ;
T = {T_0 ∪ (x^0, y^0)} ; k = 1 ;
WHILE z^U − z^L > 0 AND P^OA(T) is feasible
{
   Let (α̂, x̂, ŷ) be an optimal solution of P^OA(T) ;
   z^L = α̂ ; y^k = ŷ ;
   IF P_ŷ is feasible
   {
      Let x^k be an optimal solution of P_ŷ ;
      IF f(x^k, y^k) < z^U
      {
         z^U = f(x^k, y^k) ;
         (x^*, y^*) = (x^k, y^k) ;
      }
   }
   ELSE
   {
      Let x^k be an optimal solution of P_ŷ^V ;
   }
   T = T ∪ (x^k, y^k) ;
   k = k + 1 ;
}
```

3. Our hybrid algorithm

ALGORITHM 2: Our hybrid algorithm ;
INPUT: P: Problem (1), $OA(\bar{P}, T^I, z^U, time)$: OA procedure that address $\bar{P}$, with initial linearization points set T^I, upper bound z^U with time limited to $time$ seconds. OA procedure returns: $status$: status of OA application, $(\bar{x}, \bar{y})$: best obtained solution of $\bar{P}$, T^F: final set of linearization points, $\bar{z}^L$: lower bound to $\bar{P}$;
OUTPUT: (x^*, y^*): optimal solution of P ;

$z^U = \infty$; Let (x^0, y^0) be an optimal solution of $\tilde{P}$;
$T^P = (x^0, y^0)$ \\ Initial linearization points to P ;
$[status, (\bar{x}, \bar{y}), T^F, \bar{z}^L] = OA(\tilde{P}, T^P, z^U, OA\_time)$;
IF $status$ = "optimal solution" OR $status$ = "feasible solution"
{
$(x^*, y^*) = (\bar{x}, \bar{y})$; $z^U = f(\bar{x}, \bar{y})$;
}
IF $status$ = "optimal solution", THEN RETURN ;
Choose a variable y_j with fractional value y_j^0 ;
$Y^1 = Y \cap \{y \in \mathbb{R}^{n_y} : y_j \leq \lfloor y_j \rfloor\}$; $Y^2 = Y \cap \{y \in \mathbb{R}^{n_y} : y_j \geq \lceil y_j \rceil\}$;
Let L^i be a lower bound to node i ; $L^1 = L^2 = \max\{f(x^0, y^0), \bar{z}^L\}$;
Let $N = \{1, 2\}$ be the initial list of open nodes ;
$i = 2$; $iter = 0$; $T^P = T^P \cup T^F$;
BBLOOP:
WHILE $N \neq \emptyset$
{
Choose a node k of N ; $N = N \setminus \{k\}$; $iter = iter + 1$;
Let (x^k, y^k) be an optimal solution of $\tilde{P}_{Y^k}$;
IF $f(x^k, y^k) < z^U$
{
IF y^k is integer
{
$z^U = f(x^k, y^k)$; $(x^*, y^*) = (x^k, y^k)$; $T^P = T^P \cup \{(x^k, y^k)\}$;
$N = N \setminus \{j : L^j \geq z^U\}$); \\ Pruning
}
ELSE
{
$\bar{L} = f(x^k, y^k)$;
IF $iter \equiv 0 \pmod{freq\_OA\_subprob}$ \\ Applying OA to subproblem
{
$T^S = \{(x^k, y^k)\}$;
$[status, (\bar{x}, \bar{y}), T^F, \bar{z}^L] = OA(P_{Y^k}, T^S, z^U, OA\_time)$;
IF $status$ = "optimal solution" OR $status$ = "feasible solution"
{
$z^U = f(\bar{x}, \bar{y})$; $(x^*, y^*) = (\bar{x}, \bar{y})$; $N = N \setminus \{j : L^j \geq z^U\}$ \\ Pruning
$T^P = T^P \cup \{(\bar{x}, \bar{y})\}$;
}
IF $status$ = "optimal solution" OR $status$ = "infeasible problem"
{
GO TO BBLOOP;
}
$\bar{L} = \max\{\bar{L}, \bar{z}^L\}$;
}
Choose a variable y_j with fractional value y_j^k ; \\ Branching
$Y^{i+1} = Y^k \cap \{y \in \mathbb{R}^{n_y} : y_j \leq \lfloor y_j \rfloor\}$; $Y^{i+2} = Y^k \cap \{y \in \mathbb{R}^{n_y} : y_j \geq \lceil y_j \rceil\}$;
$L^{i+1} = L^{i+2} = \bar{L}$; $N = N \cup \{i+1, i+2\}$; $i = i + 2$;
}
}
IF $iter \equiv 0 \pmod{freq\_OA\_prob}$
{
$[status, (\bar{x}, \bar{y}), T^F, \bar{z}^L] = OA(P, T^P, z^U, OA\_time)$; \\Applying OA to P
IF $status$ = "optimal solution"
{
$(x^*, y^*) = (\bar{x}, \bar{y})$; RETURN ;
}
IF $status$ = "feasible solution"
{
$z^U = f(\bar{x}, \bar{y})$; $(x^*, y^*) = (\bar{x}, \bar{y})$; $N = N \setminus \{j : L^j \geq z^U\}$; \\ Pruning
}
$T^P = T^P \cup T^F$; \\ Accumulating linearization points
}
}

Here, we propose a new hybrid algorithm combining branch-and-bound and outer approximation algorithms, showed in Algorithm 2. The inspiration to develop this algorithm comes from the hybrid algorithm proposed by Bonami et al. in [1]. Let us first define the subproblem addressed at each node of branch-and-bound tree in a given partition $\bar{Y} \subset Y$ as:

$$
\begin{array}{lll}
(P_{\bar{Y}}) & \text{minimize}_{x,y} & f(x,y) \\
 & \text{s. t.:} & g(x,y) \leq 0 \\
 & & x \in X, \; y \in \bar{Y} \cap \mathbb{Z}^{n_y}.
\end{array}
\tag{5}
$$

The general idea behind the proposed approach is very simple: it makes space partitioning in the NLP branch-and-bound tree, and, then, applies outer approximation algorithm to some of the partitions Y^k, i.e., applies OA to some subproblems P_{Y^k}, with a time limit to spend. Bonami et al. adopt this strategy only once in their algorithm before beginning the space partitioning, i.e., OA is applied to solve the original MINLP problem in the root node, as they use the OA based branch-and-cut [5]. Here, we adopt this strategy in the root node and also in some generated subproblems along the evolution of the algorithm, resulting in several calls to OA procedure. During enumeration, the proposed algorithm comes back to the original MINLP problem considered in the root node to make new OA iterations with limited time. Integer solutions found in the branch-and-bound tree are used as linearization points when we apply again OA algorithm to solve P. On the other hand, OA algorithm collaborates with enumeration scheme providing stronger lower bounds to the addressed subtrees and integer solutions that improve the upper bound to P.

At every $freq\_OA\_subprob$ branch-and-bound iterations (e.g. 50), OA algorithm is applied to the current subproblem P_{Y^k} and at every $freq\_OA\_prob$ iterations (e.g. 200), OA algorithm is applied to the original problem P. We observe that if we interrupt the OA algorithm at the end of a given iteration, saving the set of linearization points, and later restart it by using this same set as input, OA algorithm continues as the same way as if it has never been interrupted, i.e., when we save the set of the current linearization points at the end of an iteration, we are saving the current state of algorithm as a whole. In this way, considering the calls to OA procedure at every $freq\_OA\_prob$ branch-and-bound iterations, this would be, in principle, like solving P using OA algorithm by making "pauses" in its execution. During these "pauses", the proposed algorithm performs $freq\_OA\_prob$ branch-and-bound iterations. Actually, it does not happen precisely in this way because during these "pauses", we can add new linearization points from integer solutions found by branch-and-bound algorithm. In this sense, we hope to speed up the performance of OA algorithm, i.e., saving some of its regular iterations.

The results of preliminary computational tests show that the proposed hybrid algorithm has better performance in comparison to pure outer approximation, pure branch-and-bound and hybrid from [1] algorithms on several instances of MINLP. The proposed hybrid algorithm will be available in the new open MINLP solver under development called Muriqui.

References

[1] P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, A. Wächter. An Algorithmic Framework for Convex Mixed Integer Nonlinear Programs. Discrete Optimization 5, pages 186–204, 2008.

[2] M. Duran, I. E. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Mathematical Programming 36, pages 307–339, 1986.

[3] I. E. Grossmann. Review of nonlinear mixed-integer and disjunctive programming techniques. Optimization and Engineering 3, pages 227–252, 2002.

[4] W. Melo. Algorithms for Mixed Integer Nonlinear Programming. Master thesis, Universidade Federal do Rio de Janeiro (Federal University of Rio de Janeiro), 2012 (printed in portuguese).

[5] I. Quesada, I.E. Grossmann. An LP/NLP based branch and bound algorithm for convex MINLP optimization problems. Computers & Chemical Engineering 16 10-11, pages 937–947, 1992.