A Logic-Based Implementation of the LP/NLP Branch and Bound Algorithm

Rubén Ruiz-Femeniaa\*, Juan Javaloyes-Antóna and José A. Caballeroa.

aDepartment of Chemical Engineering, University of Alicante, Ap. correos 99. E-03080, Alicante, Spain.

ruben.ruiz@ua.es

Abstract

This study addresses discrete optimization problems in Process System Engineering (PSE) by introducing a Logic-Based LP/NLP Branch and Bound (LB LP/NLP BB) algorithm. We extend the LP/NLP BB algorithm to exploit disjunctive structures, enhancing its capabilities for discrete/continuous optimization. The proposed logic-based approach reduces combinatorial search efforts, showcasing superior performance compared to non-logic versions. Through a case study and computational results, we demonstrate a substantial reduction in both the number and size of LP problems solved in the logic version. The Logic-Based LP/NLP BB algorithm emerges as a promising tool for tackling large-scale disjunctive problems, offering potential applications superstructure optimization embedded with commercial process simulators.

**Keywords**: mathematical programming, generalized disjunctive programming, LP/NLP branch and bound, Mixed-Integer Non-Linear Programming (MINLP).

* 1. Introduction

The contributions of the Process System Engineering community to solving the complex problems emerging nowadays demands addressing large-scale discrete problems. Advancements in computer hardware and mathematical programming algorithms, as noted by Koch et al. (2022), show approximately a 20x speedup in hardware for LP/MILP (Linear Programming/Mixed-Integer Linear Programming) problems from 2001 to 2020, while MILP algorithms improved 50x, resulting in a total speed up of around 1,000 times. Continuing this growth, innovative algorithms may play an essential role in solving real-world problems.

In Mixed-Integer Non-Linear Programming (MINLP), two main approaches exist: i) single-tree and ii) multi-tree search. A well-known single-tree algorithm for solving MILPs is the Kelley’s algorithm (Kelley, 1960), a branch-and-bound type algorithm that solves relaces MILP problems, where the feasible region is iteratively tightened by adding cutting planes derived from the fractional values of the solution in the previous iteration. Westerlund and Petterson extended this as the Extended Cutting Plane (ECP) algorithm (Westerlund and Pettersson, 1995). The main drawback of single-tree algorithms is that its convergence may be slow. An algorithm that exemplifies the multi-tree approach is the Outer Approximation (OA) algorithm (Duran and Grossmann, 1986), which solves the Non-Linear Programming (NLP) subproblem that arises from the original problem by fixing binary variables from the solution of the previous MILP. This significantly reduces the number of iterations compared to branch and bound algorithms (single-tree). However, it is less efficient because, at each iteration, an MILP must be solved, leading to the development of a new search tree. To leverage the strengths of both approaches, a hybrid algorithm, LP/NLP based branch and bound, was developed by Quesada and Grossmann (1992), that can be viewed as a single-tree implementation of the OA algorithm. The idea here is to relax the nonlinearities by linearizing the original problem and hence solve LP problems at the nodes of the tress while simultaneously relaxing the integrality by branching. This avoids solving an MILP master problem at each iteration. When an integer feasible solution is obtained at a node, an NLP subproblem is solved, providing an upper bound and outer approximation cuts that are updated to tighten all the open nodes of the single-tree search.

In this work, we extended the LP/NLP BB algorithm developed by Quesada and Grossmann into a logic equivalent customized algorithm that exploits the disjunctive structure of the model, thereby facilitating the modeling of discrete/continuous optimization problems by using symbolic expressions. The proposed LB LP/NLP BB algorithm solves the NLP subproblems within a reduced space (focusing exclusively on the Boolean variables that hold true in the current node). We assess the performance of the logic-based version by comparing it to the original LP/NLP algorithm through the use of a case study.

* 1. Example Logic-Based LP/NLP Brach and Bound Algorithm
     1. Types of problems

In this section we formulate the problems involved in the LB LP/NLP BB algorithm. The problem P-GDP shows the Generalized Disjunctive Programming formulation (Raman and Grossmann, 1994) of an optimization problem involving discrete decisions (given a set of disjunctions  , at each disjunction , only one term  must be selected).

|  |  |
| --- | --- |
|  | (P-GDP) |

From the above formulation, a GDP master problem, Eq (1), and GDP subproblem, Eq (2) can be derived. For convex problems, the GDP master problem is a relaxation of the original GDP problem generated by linear approximations at a set of given points, .

|  |  |
| --- | --- |
|  | (1) |

The GDP subproblem (Eq (2)), which is an NLP, is obtained from the GDP formulation by fixing the values of the Boolean variables.

|  |  |
| --- | --- |
|  | (2) |

The GDP master problem, Eq (1), is reformulated using the Hull Relaxation to obtain an MILP master problem, Eq (3):

|  |  |
| --- | --- |
|  | (3) |

* + 1. Algorithm

The Logic-Based LP/NLP BB algorithm shown in Algorithm 1 keeps a list of the  problems obtained from the GDP master HR reformulation, Eq. (3), by relaxing the integrality condition on the binary variables. Each  problem represents a node within the branch-and-bound tree. The formulation of the  at the initial node, , requires an initialization step that provides, at least, a linearization point () for the nonlinear constraints for each term for all the disjunctions. These points, , comes from the solution of  initial GDP subproblems given by Eq. (2), where the fixed values of the Boolean variables, , are determined by solving an iterative set covering problem. Let denote the list of nodes that must still be solved (i.e., those not pruned or branched). Let  and denote the best upper and lower bound on the optimum value. Initially, the upper bound is derived from the lowest value of the objective function in the solved initial GDP subproblems. The proposed method is summarized as a pseudo code in Algorithm 1.

Algorithm 1. Logic-Based LP/NLP Branch and Bound algorithm.

|  |
| --- |
| **0. Initialize**  ,, , |
| **1. Terminate?**  If , the solution is optimal or the optimality gap  is below a specified tolerance. |
| **2. Select node**  Choose a node in  and **delete** it from . |
| **3. Bound**  Solve . If it is infeasible, go to **Step 1**. Else, let be its solution and  its objective function value. |
| **4. Prune**  If , go to **Step 1**.  Else,  if  is infeasible to MILP (GDP master problem, Eq. ) go to **Step 6**.  if is feasible to MILP, let  if , and go to **Step 5.** |
| **5.** **Add cuts?**  Solve the NLP (GDP subproblem, Eq. ) fixing the Boolean variables accordingly to  values. If , let ,  and **delete** from  all nodes  with . Strengthen and all nodes  in by adding linearizations evaluated at the solution of the NLP, . Go back to **Step 3**. |
| **6. Branch**  From , construct linear programs  and with smaller feasible regions whose union contains all the solutions of  with . **Add** the corresponding new nodes  and to and go to **Step 1**. |

* 1. Case study

The proposed Logic-Based LP/NLP branch and bound algorithm has been tested in solving the illustrative eight-process problem taken from the work of Türkay and Grossmann (1996). This test problem comprises eigth Boolean variables, 33 continuous variables, and eight disjunctions with two terms each (five of them nonlinear), leading to 20 different feasible process topologies. The superstructure of the process syntesis problem is shown in Figure 1a.

* 1. Results

The case study is solved by the LB LP/NLP BB algorithm implemented in GAMS using the CONOPT solver for the NLP subproblems and the CPLEX 12.1.0 solver for LP problems. Figure 2 shows the search tree for the case study obtained with the proposed method. It follows a best-bound search strategy and a branching rule that selects the binary variable closest to 1. The optimal process configuration for the test problem is shown in Figure 1b. To assess the performance of the LP LP/NLP BB algorithm, the test problem has also been solved with the non-logic version of the LP/NLP BB algorithm and the Logic-Based Outer Approximation (LBOA) algorithm. Table 1 presents the main computational results. The number of LP problems solved decreased from 32 to 10 in the logic version of the LP/NLP BB algorithm. In comparison to the LBOA, the proposed method eliminates the need to solve 3 MILP problems.

Regarding the size of the LP problems, a typical LP problem in the logic version of the LP/NLP BB algorithm comprises 56 equations, whereas in the non-logic algorithm, it consists of 75 equations. In both cases, there are 33 continuous variables and 8 relaxed binary variables.

* 1. Conclusions

The application of logic version of the LP/NLP BB algorithm to our case study has resulted in two main advantages over the conventional LP/NLP BB. It achieves a significant descent in the number of LP problems solved and in the size of these problems. These results are promising when applied to larger-scale disjunctive problem. The logic feature of the LB LP/NLP BB algorithm is perfectly suited for superstructure optimization involving the use of commercial process simulators.

Further work involves adapting the LB LP/NLP BB algorithm to be used in conjunction with the state-of-the-art branch and bound commercial solvers through the user cuts facility available in these solvers.

|  |  |
| --- | --- |
|  |  |
| a) | b) |

Figure 1. a) Eight-process problem superstructure and b) optimal flowsheet topology.



Figure 2. Search tree of the Logic-Based LP/NLP branch and bound algorithm.

Table 1. Computational results.

|  |  |  |  |
| --- | --- | --- | --- |
|  | LB LP/NLP BB | LBOA | LP/NLP BB |
| Initial NLP sub-problems | 2 | 1 | 1 |
| NLP sub-problems | 2 | 2 | 3 |
| LP problems | 10 | --- | 32 |
| MILP problems | --- | 3 | --- |

Acknowledgments

The authors gratefully acknowledge financial support to the Spanish “Ministerio de Ciencia e Innovación” under project PID2021-124139NB-C21.

References

Duran, M.A., Grossmann, I.E., 1986. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Math. Program. 36, 307–339.

J. E. Kelley, J., 1960. The Cutting-Plane Method for Solving Convex Programs. J. Soc. Ind. Appl. Math. 8, 703–712.

Koch, T., Berthold, T., Pedersen, J., Vanaret, C., 2022. Progress in mathematical programming solvers from 2001 to 2020. EURO J. Comput. Optim. 10, 100031.

Quesada, I., Grossmann, I.E., 1992. An LP/NLP based branch and bound algorithm for convex MINLP optimization problems. Comput. Chem. Eng. 16, 937–947.

Raman, R., Grossmann, I.E., 1994. Modelling and computational techniques for logic based integer programming. Comput. Chem. Eng. 18, 563–578.

Türkay, M., Grossmann, I.E., 1996. Logic-based MINLP algorithms for the optimal synthesis of process networks. Comput. Chem. Eng. 20, 959–978.

Westerlund, T., Pettersson, F., 1995. An extended cutting plane method for solving convex MINLP problems. Comput. Chem. Eng. 19, 131–136.