GENERALIZED DISJUNCTIVE PROGRAMMING: A FRAMEWORK FOR FORMULATION AND ALTERNATIVE ALGORITHMS FOR MINLP OPTIMIZATION

IGNACIO E. GROSSMANN* AND JUAN P. RUIZ†

Abstract.
Generalized disjunctive programming (GDP) is an extension of the disjunctive programming paradigm developed by Balas. The GDP formulation involves Boolean and continuous variables that are specified in algebraic constraints, disjunctions and logic propositions, which is an alternative representation to the traditional algebraic mixed-integer programming formulation. After providing a brief review of MINLP optimization, we present an overview of GDP for the case of convex functions emphasizing the quality of continuous relaxations of alternative reformulations that include the big-M and the hull relaxation. We then review disjunctive branch and bound as well as logic-based decomposition methods that circumvent some of the limitations in traditional MINLP optimization. We next consider the case of linear GDP problems to show how a hierarchy of relaxations can be developed by performing sequential intersection of disjunctions. Finally, for the case when the GDP problem involves nonconvex functions, we propose a scheme for tightening the lower bounds for obtaining the global optimum using a combined disjunctive and spatial branch and bound search. We illustrate the application of the theoretical concepts and algorithms on several engineering and OR problems.

Key words. Disjunctive Programming, Mixed Integer Non-Linear Programming, Global Optimization

AMS(MOS) subject classifications.

1. Introduction.

Mixed-integer optimization provides a framework for mathematically modeling many optimization problems that involve discrete and continuous variables. Over the last few years there has been a pronounced increase in the development of these models, particularly in process systems engineering (see Grossmann et al, 1999; Kallrath, 2000; Mendez et al, 2006).

Mixed-integer linear programming (MILP) methods and codes such as CPLEX, XPRESS and GUROBI have made great advances and are currently applied to increasingly larger problems. Mixed-integer nonlinear programming (MINLP) has also made significant progress as a number of codes have been developed over the last decade (e.g. DICOPT, SBB, a-ECP, Bonmin, FilMINT, BARON, etc.). Despite these advances, three basic questions still remain in this area: a) How to develop the "best" model?, b) How to improve the relaxation in these models?, c) How to solve nonconvex GDP problems to global optimality?

Motivated by the above questions, one of the trends has been to represent

*Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, grossmann@cmu.edu, NSF OCI-0750826
†

discrete/continuous optimization problems by models consisting of algebraic constraints, logic disjunctions and logic relations (Raman and Grossmann, 1994; Hooker and Osorio, 1999). The basic motivation in using these representations is: a) to facilitate the modeling of discrete/continuous optimization problems, b) to retain and exploit the inherent logic structure of problems to reduce the combinatorics and to improve the relaxations, and c) to improve the bounds of the global optimum in nonconvex problems. In this paper we provide an overview of Generalized Disjunctive Programming (Raman and Grossmann, 1994), which can be regarded as a generalization of disjunctive programming (Balas, 1985). In contrast to the traditional algebraic mixed-integer programming formulations, the GDP formulation involves Boolean and continuous variables that are specified in algebraic constraints, disjunctions and logic propositions. After providing a brief review of MINLP optimization, we address the solution of GDP problems for the case of convex functions for which we consider the big-M and the hull relaxation MINLP reformulations. We then review disjunctive branch and bound as well as logic-based decomposition methods that circumvent some of the MINLP reformulations. We next consider the case of linear GDP problems to show how a hierarchy of relaxations can be developed by performing sequential intersection of disjunctions. Finally, for the case when the GDP problem involves nonconvex functions, we describe a scheme for tightening the lower bounds for obtaining the global optimum using a combined disjunctive and spatial branch and bound search. We illustrate the application of the theoretical concepts and algorithms on several engineering and OR problems.

2. Review of MINLP Methods. Since GDP problems are often reformulated as algebraic MINLP problems we provide a brief review of these methods. The most basic form of an MINLP problem is as follows:

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

where $f : R^n \to R^1, g : R^n \to R^m$ are *differentiable* functions, J is the index set of constraints, and x and y are the continuous and discrete variables, respectively. In the general case the MINLP problem will also involve nonlinear equations, which we omit here for convenience in the presentation. The set X commonly corresponds to a convex compact set, e.g. $X = \{x | x \in R^n, Dx \leq d, x^L \leq x \leq x^U\}$; the discrete set Y corresponds to a polyhedral set of integer points, $Y = \{y | y \in Z^m, Ay \leq a\}$, which in most applications is restricted to 0-1 values, $y \in \{0, 1\}^m$. In most applications of interest the objective and constraint functions *f(), g()* are linear in y (e.g. fixed cost charges and mixed-logic constraints): $f(x,y) = c^T y + r(x)$, $g(x,y) = By + h(x)$.The derivation of most methods for MINLP assumes that the functions f and g are convex.

Methods that have addressed the solution of problem (MINLP) include the branch and bound method (BB) (Gupta and Ravindran, 1985; Borchers and Mitchell, 1994; Stubbs and Mehrotra, 1999; Leyffer, 2001), Generalized Benders Decomposition (GBD) (Geoffrion, 1972), Outer-Approximation (OA) (Duran and Grossmann, 1986; Yuan et al., 1988; Fletcher and Leyffer, 1994), LP/NLP based branch and bound (Quesada and Grossmann, 1992; Bonami et al, 2008), and Extended Cutting Plane Method (ECP) (Westerlund and Pettersson, 1995, and Westerlund and Porn (2002)).
As discussed in Grossmann (2002) these algorithms can be classified in terms of the following basic subproblems that are involved in these methods:
NLP Subproblems.
a) NLP relaxation

$$
\begin{array}{rl}
 & \min \; Z_{LB}^k = f(x,y) \\
s.t. & g_j(x,y) \leq 0 \quad j \in J \\
 & x \in X, \quad y \in Y_R \\
 & y_i \leq \alpha_i^k \quad i \in I_{FL}^k \\
 & y_i \geq \beta_i^k \quad i \in I_{FU}^k
\end{array}
\qquad \text{(NLP1)}
$$

where Y_R is the continuous relaxation of the set Y, and I_{FL}^k, I_{FU}^k are index subsets of the integer variables $y_i, i \in I$, which are restricted to lower and upper bounds, at the *k'th* step of a branch and bound enumeration procedure. If $I_{FU}^k = I_{FL}^k = \emptyset$ *(k=0)*, (NLP1) corresponds to the continuous NLP relaxation of (P1), whose optimal objective function Z_{LB}^0 provides an absolute lower bound to (MINLP)..
b) NLP subproblem for fixed y^k:

$$
\begin{array}{rl}
 & \min \; Z_U^k = f(x,y^k) \\
s.t. & g_j(x,y^k) \leq 0 \quad j \in J \\
 & x \in X
\end{array}
\qquad \text{(NLP2)}
$$

which yields an upper bound Z_U^k to (MINLP) provided (NLP2) has a feasible solution. When this is not the case, we consider the next subproblem:
c) Feasibility subproblem for fixed y^k:

$$
\begin{array}{rl}
 & \min \; u \\
s.t. & g_j(x,y^k) \leq u \quad j \in J \\
 & x \in X, \quad u \in R^1
\end{array}
\qquad \text{(NLPF)}
$$

which can be interpreted as the minimization of the infinity-norm measure of infeasibility of the corresponding NLP subproblem. Note that for an infeasible subproblem the solution of (NLPF) yields a strictly positive value of the scalar variable u.
MILP cutting plane.
The convexity of the nonlinear functions is exploited by replacing them with supporting hyperplanes, that are generally, but not necessarily, derived

at the solution of the NLP subproblems. In particular, the new values $y^K(or\ (x^K, y^K))$ are obtained from a cutting plane MILP problem that is based on the K points, (x^k, y^k), $k = 1, 2...K$ generated at the K previous steps:

$$
\begin{array}{ll}
\min & Z_L^K = \alpha \\
st & \left.\begin{array}{l}
\alpha \ \geq \ f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \\
g_j(x^k, y^k) + \nabla g_j(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \ \leq \ 0 \ \ j \in J^k
\end{array}\right\} k = 1, ..K \\
& x \in X, \ \ y \in Y
\end{array}
\qquad \text{(M-MIP)}
$$

where $J^K \subseteq J$. When only a subset of linearizations is included, these commonly correspond to violated constraints in problem (P1). Alternatively, it is possible to include all linearizations in (M-MIP). The solution of (M-MIP) yields a valid lower bound Z_L^K to problem (MINLP), which is nondecreasing with the number of linearization points K.
The different methods can be classified according to their use of the subproblems (NLP1), (NLP2) and (NLPF), and the specific specialization of the MILP problem (M-MIP) as seen in Fig. 1.

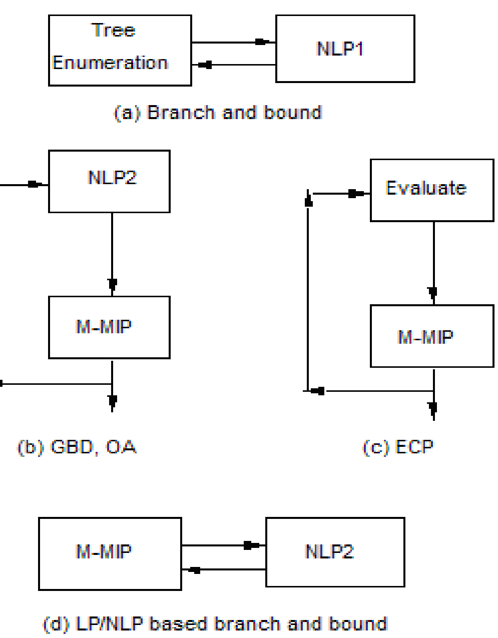

FIG. 1. *Major Steps in the different MINLP Algorithms*

The number of computer codes for solving MINLP problems has increased in the last decade. The program DICOPT (Viswanathan and Grossmann, 1990) is an MINLP solver that is available in the modeling system GAMS (Brooke et al., 1998), and is based on the outer-approximation method. For handling nonconvexities, slack variables are introduced in the master problem. Since the bounding properties cannot be guaranteed for this extension, the search for nonconvex problems is terminated when there is no further improvement in the objective of the feasible NLP subproblems, which is a heuristic that works reasonably well. A similar code to DICOPT, AAOA, is available in AIMMS. Codes that implement the branch-and-bound method using subproblems (NLP1) include the code MINLP_BB that is based on an SQP algorithm (Leyffer, 2001) and is available in AMPL, and the code SBB which is available in GAMS (Brooke et al, 1998). Both codes assume that the bounds are valid evern though the original problem may be nonconvex. The code α–ECP that is available in GAMS implements the extended cutting plane method by Westerlund and Pettersson (1995), including the extension by Westerlund and Prn (2002). The code MINOPT (Schweiger and Floudas, 1998) also implements the OA and GBD methods, and applies them to mixed-integer dynamic optimization problems. The open source code Bonmin (Bonami et al, 2008) implements the branch and bound method, the outer-approximation and an extension of the LP/NLP based branch and bound method in one single framework. FilMINT (Abhishek, Linderoth and Leyffer, 2006) also implements a variant of the the LP/NLP based branch and bound method. Codes for the global optimization that implement the spatial branch and bound method include BARON (Sahinidis, 1996), LINDOGlobal (Lindo Systems, Inc.), and Couenne (Belotti, 2009).

3. Nonlinear Generalized Disjunctive Programming. An alternative approach for representing discrete/continuous optimization problems is by using models consisting of algebraic constraints, logic disjunctions and logic propositions (Beaumont,1991; Raman and Grossmann,1994; Turkay and Grossmann, 1996; Hooker and Osorio, 1999; Hooker, 2000; Lee and Grossmann, 2000). This approach not only facilitates the development of the models by making the formulation process intuitive, but it also keeps in the model the underlying logic structure of the problem that can be exploited to find the solution more efficiently. A particular case of these models is generalized disjunctive programming (GDP) (Raman and Grossmann, 1994) the main focus of this paper, and which can be regarded as a generalization of disjunctive programming (Balas, 1985). Process Design and Planning and Scheduling are some of the areas where GDP formulations have shown to be successful.

3.1. Formulation. The general structure of a GDP can be represented as follows (Raman & Grossmann, 1994):

$$Min\ Z = f(x) + \sum_{k\in K} c_k$$

$$s.t.\ g(x) \le 0$$

$$\underset{i\in D_k}{\vee} \begin{bmatrix} Y_{ik} \\ r_{ik}(x) \le 0 \\ c_k = \gamma_{ik} \end{bmatrix} \qquad k \in K \qquad \text{(GDP)}$$

$$\Omega\ (Y)=True$$

$$x^{lo} \le x \le x^{up}$$

$$x \in R^n\ ,\ c_k \in R^1,\ Y_{ik} \in \{True, False\}$$

where $f : R^n \to R^1$ is a function of the continuous variables x in the objective function, $g : R^n \to R^l$ belongs to the set of global constraints, the disjunctions $k \in K$, are composed of a number of terms $i \in Dk$, that are connected by the OR operator . In each term there is a Boolean variable Y_{ik}, a set of inequalities $r_{ik}(x) \le 0$, $r_{ik} : R^n \to R^j$, and a cost variable c_k. If Y_{ik} is true, then $r_{ik}(x) \le 0$ and $c_k = \gamma_{ik}$ are enforced; otherwise they are ignored. Also, $\Omega(Y) = True$ are logic propositions for the Boolean variables expressed in the conjunctive normal form $\Omega(Y) = \underset{t=1,2,..T}{\wedge} \left[\underset{Y_{jk}\in R_t}{\vee} (Y_{jk}) \underset{Y_{jk}\in Q_t}{\vee} (\neg Y_{jk}) \right]$ where for each clause t, t=1,2. . . .T, R_t is the subset of Boolean variables that are non-negated, and Q_t is the subset of Boolean variables that are negated. As indicated in Sawaya & Grossmann (2008), we assume that the logic constraints $\underset{j\in J}{\veebar} Y_{ik}$ are contained in $\Omega(Y) = True$.

There are three major cases that arise in problem (GDP): **a)** linear functions f, g and r **b)** convex nonlinear functions f, g and r **c)** nonconvex functions f, g and r. Each of these cases require different solution methods.

3.2. Illustrative Example. The following example aims at illustrating how the GDP framework can be used to model the optimization of a simple process network shown in Fig 2 that produces a product B by consuming a raw material A. The variables F represent material flows. The problem is to determine the amount of product to produce (F8) with a selling price P1, the amount of raw material to buy (F1) with a cost P2 and the set of unit operations to use (i.e. HX1, R1, R2, DC1) with a cost c k $k \in \{HX1, R1, R2, DC1\}$, in order to maximize the profit.

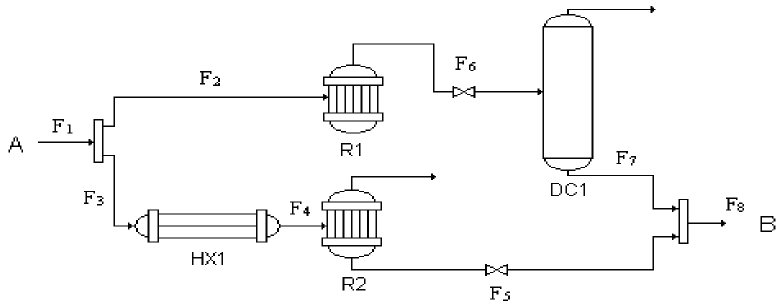

Fig. 2. *Process network example*

The generalized disjunctive program that represents the problem can be formulated as follows:

$$Max\, Z = \ P_1\, F_8 - \ P_2\, F_1 - \sum_{k \in K} c_k$$

$s.t.$

$$F_1 = F_3 + F_2 \qquad (1)$$

$$F_8 = F_7 + F_5 \qquad (2)$$

$$\begin{bmatrix} Y_{HX1} \\ F_4 = F_3 \\ c_{HX1} = \gamma_{HX1} \end{bmatrix} \vee \begin{bmatrix} \neg Y_{HX1} \\ F_4 = F_3 = 0 \\ c_{HX1} = 0 \end{bmatrix} \qquad (3)$$

$$\begin{bmatrix} Y_{R2} \\ F_5 = \beta_1 F_4 \\ c_{R2} = \gamma_{R2} \end{bmatrix} \vee \begin{bmatrix} \neg Y_{R2} \\ F_5 = F_4 = 0 \\ c_{R2} = 0 \end{bmatrix} \qquad (4)$$

$$\begin{bmatrix} Y_{R1} \\ F_6 = \beta_2 F_2 \\ c_{R1} = \gamma_{R1} \end{bmatrix} \vee \begin{bmatrix} \neg Y_{R1} \\ F_6 = F_2 = 0 \\ c_{R1} = 0 \end{bmatrix} \qquad (5)$$

$$\begin{bmatrix} Y_{DC1} \\ F_7 = \beta_3 F_6 \\ c_{DC1} = \gamma_{DC1} \end{bmatrix} \vee \begin{bmatrix} \neg Y_{DC1} \\ F_7 = F_6 = 0 \\ c_{DC1} = 0 \end{bmatrix} \qquad (6)$$

$$Y_{R2} \Leftrightarrow Y_{HX1} \qquad (7)$$

$$Y_{R1} \Leftrightarrow Y_{DC1} \qquad (8)$$

$F_i \in R\,,\ c_k \in R^1,\ Y_k \in \{True, False\} \quad i \in \{1,2,3,4,5,6,7,8\}$
$k \in \{HX1, R1, R2, DC1\}$

where (1) represents the objective function, (2) and (3) are the global constraints representing the mass balances around the splitter and mixer respectively, the disjunctions (4),(5),(6) and (7) represent the existence or non-existence of the unit operation k , $k \in \{HX1, R1, R2, DC1\}$with their respective characteristic equations and (8) and (9) the logic propositions which enforce the selection of DC1 if and only if R1 is chosen and HX1 if and only if R2 is chosen. For the sake of simplicity we have presented here a simple linear model. In the actual application to a process problem there would be thousands of nonlinear equations.

3.3. Solution Methods.

In order to take advantage of the existing MINLP solvers, GDPs are often **reformulated** as an MINLP by using either the Big-M (BM) (Nemhauser & Wolsey (1988)), or the Convex Hull (CH) (Lee & Grossmann (2000)) reformulation. The former yields:

$$Min\, Z = f(x) + \sum_{i\in D_k}\sum_{k\in K}\gamma_{ik}y_{ik}$$

$$\begin{array}{lll} s.t.\ \ g(x) \leq 0 & & \\ r_{ik}(x) \leq M(1-y_{ik}) & k \in K,\ i \in D_k & \quad (\text{BM}) \\ \sum_{i\in D_k} y_{ik} = 1 & k \in K & \\ Ay \geq a & & \end{array}$$

$$x \in R^n, y_{ik} \in \{0,1\}\ \ k \in K, i \in D_k$$

where the variable y_{ik} has a one to one correspondence with the Boolean variable Y_{ik}.
Note that when $y_{ik} = 0$ and the parameter M is large enough, the associated constraint becomes redundant; otherwise, it is enforced. Also, $Ay = a$ is the reformulation of the logic constraints in the discrete space, which can be easily accomplished as described in Williams (1985) and discussed in Raman and Grossmann (1991). The convex hull reformulation yields,

$$Min\, Z = f(x) + \sum_{i\in D_k}\sum_{k\in K}\gamma_{ik}y_{ik}$$

$$\begin{array}{ll} s.t. & \\ x = \sum_{i\in D_K} v^{ik} \quad k \in K & \\ g(x) \leq 0 & \\ y_{ik}r_{ik}(\nu^{ik}/y_{ik}) \leq 0\ \ k \in K,\ i \in D_k & \quad (\text{CH}) \\ 0 \leq \nu^{ik} \leq y_{ik}U_v\ \ k \in K,\ i \in D_k & \\ \sum_{i\in D_k} y_{ik} = 1\ \ k \in K & \\ Ay \geq a & \end{array}$$

$$x \in R^n,\ v_{ik} \in R^1, c_k \in R^1, y_{ik} \in \{0,1\}\ \ k \in K, i \in D_k$$

As it can be seen, the CH reformulation is less intuitive than the BM. However, there is also a one to one correspondence between (GDP) and (CH).
Note that the size of the problem is increased by introducing a new set of disaggregated variables ν^{ik} and new constraints. On the other hand, as proved in Grossmann and Lee (2003) and discussed by Vecchietti, Lee, Grossmann (2003), the CH formulation is at least as tight and generally tighter than the BM when the discrete domain is relaxed (i.e. $0 \leq y_{ik} \leq 1,\ k \in K, i \in D_k$) . This is of great importance considering that the efficiency of the MINLP solvers heavily rely on the quality of these relaxations.
It is important to note that on the one hand the term $y_{ik} r_{ik}(\nu^{ik}/y_{ik})$ is convex if $r_{ik}(x)$ is a convex function. On the other hand the term requires the use of a suitable approximation to avoid singularities. Sawaya & Grossmann (2007) proposed the following reformulation which yields an exact approximation at $y_{ik} = 0$and $y_{ik} = 1$ for *any* value of ϵ in the interval (0,1), and the feasibility and convexity of the approximating problem are maintained.
$y_{ik} r_{ik}(\nu^{ik}/y_{ik}) \approx\ ((1-\varepsilon)y_{ik}+\varepsilon) r_{ik}(\nu^{ik}/((1-\varepsilon)y_{ik}+\varepsilon)) - \varepsilon r_{ik}(0)(1-y_{ik})$
Note that this approximation assumes that$r_{ik}(x)$ is defined at x = 0
In order to fully exploit the logic structure of GDP problems, two other solution methods have been proposed for the case of convex nonlinear GDP, namely, the **Branch and Bound** method (Lee & Grossmann, 2000), which builds on the concept of disjunctive Branch and Bound method by Beaumont(1991) and the **Logic Based Outer Approximation** method (Turkay and Grossmann, 1996).
The basic idea in the **B&B method** is to directly branch on the constraints corresponding to particular terms in the disjunctions, while considering the convex hull of the remaining disjunctions. Although the tightness of the relaxation at each node is comparable with the one obtained when solving the CH reformulation with a MINLP solver (as described in section 2), the size of the problems solved are smaller and the numerical robustness is improved.
For the case of **Logic Based Outer Approximation** methods, similar to the case of OA for MINLP, the main idea is to solve iteratively a Master problem given by a Linear GDP, which will give a lower bound of the solution and an NLP subproblem that will give an upper bound. As described in Turkay and Grossmann (1996), for fixed values of the Boolean Variables, $Y_{\hat{i}k} = true$, $Y_{ik} = false$ with $\hat{i} \neq i$, the corresponding NLP subproblem (SNLP) is as follows:

$$
\begin{aligned}
&Min\ Z = f(x) + \textstyle\sum_{k \in K} c_k \\
&s.t.\ g(x) \leq 0 \\
&\left.\begin{aligned} r_{ik}(x) \leq 0 \\ c_k = \gamma_{ik} \end{aligned}\right\} for\, Y_{ik} = true\ \ i \in D_k,\, k \in K \qquad \text{(SNLP)}
\end{aligned}
$$

$x^{lo} \leq x \leq x^{up}$
$x \in R^n$, $c_k \in R^1$, $Y_{ik} \in \{True, False\}$

It is important to note that only the constraints that belong to the active terms in the disjunction (i.e. associated Boolean variable $Y_{ik} = True$) are imposed. This leads to a substantial reduction in the size of the problem compared to the direct application of the traditional AO method on the MINLP reformulation (as described in section 2). Assuming that L subproblems are solved in which sets of linearizations $\ell = 1, 2....L$ are generated for subsets of disjunction terms $L_{ik} = \{\ell | Y_{ik}^{\ell} = True\}$, one can define the following disjunctive OA master problem (MLGDP):

$$
\begin{aligned}
& Min\ Z = \alpha + \textstyle\sum_{k \in K} c_k \\
& s.t. \\
& \left.\begin{aligned} \alpha \geq f(x^{\ell}) + \nabla f(x^{\ell})^T (x - x^{\ell}) \\ g(x^{\ell}) + \nabla g(x^{\ell})^T (x - x^{\ell}) \leq 0 \end{aligned}\right\} \ \ell = 1, 2....., L \\
& \bigvee_{i \in D_k} \begin{bmatrix} Y_{ik} \\ r_{ik}(x^{\ell}) + \nabla r_{ik}(x^{\ell})(x - x^{\ell}) \leq 0 \quad \ell \in L_{ik} \\ c_k = \gamma_{ik} \end{bmatrix} k \in K \qquad \text{(MLGDP)} \\
& \Omega(Y) = True \\
& x^{lo} \leq x \leq x^{up} \\
& \alpha \in R^1,\ x \in R^n\ ,\ c_k \in R^1,\ Y_{ik} \in \{True, False\}
\end{aligned}
$$

It should be noted that before applying the above master problem is necessary to solve various subproblems (SNLP) for different values of the Boolean Variables Y_{ik} so as to produce one linear approximation of each of the terms $i \in D_k$ in the disjunctions $k \in K$. As shown by Turkay and Grossmann (1996) selecting the smallest number of subproblems amounts to solving a set covering problem, which is of small size and easy to solve. It is important to note that the number of subproblems solved in the initialization is often small since the combinatorial explosion that one might expect is in general limited by the propositional logic. Moreover, terms in the disjunctions that contain only linear functions are not necessary to be considered for generating the subproblems. This frequently arises in Process Networks since they are often modeled by using two terms disjunctions where one of the terms is always linear (see remark below). Also, it should be noted that the master problem can be reformulated as an MILP by using the big-M or Convex Hull reformulation, or else solved directly with a disjunctive branch and bound method.

Remark

In the context of process networks the disjunctions in (GDP) typically arise for each unit i in the following form:

$$\begin{bmatrix} Y_i \\ r_i(x) \le 0 \\ c_i = \gamma_i \end{bmatrix} \vee \begin{bmatrix} \neg Y_i \\ B^i x = 0 \\ c_i = 0 \end{bmatrix} \quad i \in I$$

in which the inequalities r_i apply and a fixed cost g_i is incurred if the unit is selected (Y_i); otherwise ($\neg Y_i$) there is no fixed cost and a subset of the x variables is set to zero.

3.3.1. Example. We present here numerical results on an example problem dealing with the synthesis of a process network that was originally formulated by Duran and Grossmann (1986) as an MINLP problem, and later by Turkay and Grossmann (1986) as a GDP problem. Fig. 3 shows the superstructure that involves the possible selection of 8 processes. The Boolean variables Y_j denote the existence or non-existence of processes 1-8. The global optimal solution is Z*=68.01, consists of the selection of processes 2,4,6 and 8

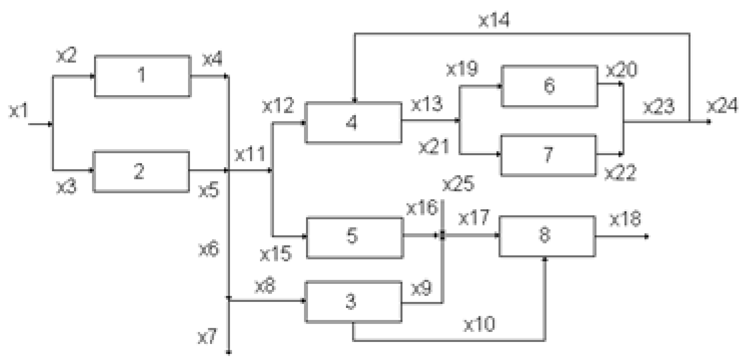

FIG. 3. *Superstructure for Process Network*

The model in the form of the GDP problem involves disjunctions for the selection of units, and propositional logic for the relationship of these units. Each disjunction contains the equation for each unit (these relax as convex inequalities). The model is as follows:

Objective function:

$Min\, Z = c_1 + c_2 + c_3 + c_4 + c_5 + c_6 + c_7 + c_8 + x_2 - 10x_3 + x_4 - 15x_5 - 40x_9 + 15x_{10} + 15x_{14} + 80x_{17} - 65x_{18} + 25x_9 - 65x_{18} + 25x_{19} - 60x_{20} + 35x_{21} - 80x_{22} - 35x_{25} + 122$

Material balances at mixing/splitting points:

$x_3 + x_5 - x_6 - x_{11} = 0$
$x_{13} - x_{19} - x_{21} = 0$
$x_{17} - x_9 - x_{16} - x_{25} = 0$
$x_{11} - x_{12} - x_{15} = 0$
$x_6 - x_7 - x_8 = 0$
$x_{23} - x_{20} - x_{22} = 0$
$x_{23} - x_{14} - x_{24} = 0$

Specifications on the flows:

$x_{10} - 0.8x_{17} \leq 0$
$x_{10} - 0.4x_{17} \geq 0$
$x_{12} - 5x_{14} \leq 0$
$x_{12} - 2x_{14} \geq 0$

Disjunctions:

$$\text{Unit 1:} \quad \begin{bmatrix} Y_1 \\ e^{x_3} - 1 - x_2 \leq 0 \\ c_1 = 5 \end{bmatrix} \vee \begin{bmatrix} \neg Y_1 \\ x_2 = x_3 = 0 \\ c_1 = 0 \end{bmatrix}$$

$$\text{Unit 2:} \quad \begin{bmatrix} Y_2 \\ e^{x_5/1.2} - 1 - x_4 \leq 0 \\ c_2 = 8 \end{bmatrix} \vee \begin{bmatrix} \neg Y_2 \\ x_4 = x_5 = 0 \\ c_2 = 0 \end{bmatrix}$$

$$\text{Unit 3:} \quad \begin{bmatrix} Y_3 \\ 1.5x_9 - x_8 + x_{10} \leq 0 \\ c_3 = 6 \end{bmatrix} \vee \begin{bmatrix} \neg Y_3 \\ x_8 = x_9 = x_{10} = 0 \\ c_3 = 0 \end{bmatrix}$$

$$\text{Unit 4:} \quad \begin{bmatrix} Y_4 \\ 1.5(x_{12} + x_{14}) - x_{13} = 0 \\ c_4 = 10 \end{bmatrix} \vee \begin{bmatrix} \neg Y_4 \\ x_{12} = x_{13} = x_{14} = 0 \\ c_4 = 0 \end{bmatrix}$$

$$\text{Unit 5:} \quad \begin{bmatrix} Y_5 \\ x_{15} - 2x_{16} = 0 \\ c_5 = 6 \end{bmatrix} \vee \begin{bmatrix} \neg Y_5 \\ x_{15} = x_{16} = 0 \\ c_5 = 0 \end{bmatrix}$$

$$\text{Unit 6:} \quad \begin{bmatrix} Y_6 \\ e^{x_{20}/1.5} - 1 - x_{19} = 0 \\ c_6 = 7 \end{bmatrix} \vee \begin{bmatrix} \neg Y_6 \\ x_{19} = x_{20} = 0 \\ c_6 = 0 \end{bmatrix}$$

$$\text{Unit 7:} \quad \begin{bmatrix} Y_7 \\ e^{x_{22}} - 1 - x_{21} = 0 \\ c_7 = 4 \end{bmatrix} \vee \begin{bmatrix} \neg Y_7 \\ x_{21} = x_{22} = 0 \\ c_7 = 0 \end{bmatrix}$$

$$\text{Unit 8:} \quad \begin{bmatrix} Y_8 \\ e^{x_{18}} - 1 - x_{10} - x_{17} = 0 \\ c_8 = 5 \end{bmatrix} \vee \begin{bmatrix} \neg Y_8 \\ x_{10} = x_{17} = x_{18} = 0 \\ c_8 = 0 \end{bmatrix}$$

Propositional Logic

$Y_1 \Rightarrow Y_3 \vee Y_4 \vee Y_5; Y_2 \Rightarrow Y_3 \vee Y_4 \vee Y_5; Y_3 \Rightarrow Y_1 \vee Y_2; Y_3 \Rightarrow Y_8$
$Y_4 \Rightarrow Y_1 \vee Y_2; Y_4 \Rightarrow Y_6 \vee Y_7; Y_5 \Rightarrow Y_1 \vee Y_2; Y_5 \Rightarrow Y_8$
$Y_6 \Rightarrow Y_4; Y_7 \Rightarrow Y_4; Y_5 \Rightarrow Y_8; Y_6 \Rightarrow Y_4; Y_7 \Rightarrow Y_4$
$Y_8 \Rightarrow Y_3 \vee Y_5 \vee (\neg Y_3 \wedge \neg Y_5)$

Specifications

$Y_1 \veebar Y_2; Y_4 \veebar Y_5; Y_6 \veebar Y_7$

Variables

$x_j,\, c_i \geq 0, Y_i = \{True, False\}\ i = 1, 2...8,\ j = 1, 2.....25$

The following Table 1 shows a comparison between the three solution approaches presented before. Master and NLP represent the number of master problems and NLP subproblems solved to find the solution. It should be noted that the Logic-Based Outer-Approximation method required solving only 3 NLP subproblems to initialize the master problem (MGDLP), which was reformulated as an MILP using the convex hull reformulation.

Table 1: GDP Solution Methods Results

	Outer Approximation*	**B&B**	**Logic Based OA ****
NLP	2	5	4
Master	2	0	1

* Solved with DICOPT through EMP (GAMS) **Solved with LOGMIP (GAMS)

3.4. Linear Generalized Disjunctive Programming. A particular class of GDP problems arises when the functions in the objective and constraints are linear. The general formulation of a Linear GDP as described by Raman and Grossmann (1994) is as follows:

$$\begin{aligned}
&Min\, Z = d^T x + \textstyle\sum_k c_k \\
&s.t.\ \ Bx \leq b \\
&\bigvee_{i \in D_k} \begin{bmatrix} Y_{ik} \\ A_{ik} x \leq a_{ik} \\ c_k = \gamma_{ik} \end{bmatrix} k \in K \qquad \text{(LGDP)} \\
&\Omega(Y) = True \\
&x^{lo} \leq x \leq x^{up}
\end{aligned}$$

$x \in R^n,\ c_k \in R^1,\ Y_{ik} \in \{True, False\}\,,\ k \in K\ ,\ i \in D_k$

The big-M formulation reads:

$$Min\, Z = d^T x + \sum_{i \in D_k} \sum_{k \in K} \gamma_{ij} y_{ik}$$

$$\begin{aligned}
&s.t.\ \ Bx \leq b \\
&A_{ik}x \leq a_{ik} + M(1 - y_{ik}) \qquad k \in K,\ i \in D_k \qquad \text{(LBM)} \\
&\textstyle\sum_{i \in D_k} y_{ik} = 1 \quad k \in K \\
&Ay \geq a
\end{aligned}$$

$$x \in R^n, y_{ik} \in \{0, 1\} \ \ k \in K, i \in D_k$$

while the CH formulation reads:

$$Min\, Z = d^T x + \sum_{i \in D_k} \sum_{k \in K} \gamma_{ij} y_{ik}$$
$s.t.$

$$\begin{aligned}
&x = \textstyle\sum_{i \in D_K} v^{ik} \qquad k \in K \\
&Bx \leq b \\
&A_{ik}\nu^{ik} \leq a_{ik} y_{ik} \qquad k \in K,\ i \in D_k \qquad \text{(LCH)} \\
&0 \leq \nu^{ik} \leq y_{ik} U_v \quad k \in K,\ i \in D_k \\
&\textstyle\sum_{i \in D_k} y_{ik} = 1 \ \ k \in K \\
&Ay \geq a
\end{aligned}$$

$$x \in R^n,\ v_{ik} \in R^1, c_k \in R^1, y_{ik} \in \{0, 1\} \ \ k \in K, i \in D_k$$

As a particular case of a GDP, LGDPs can be solved using MIP solvers applied on the LBM or LCH reformulations. However, as described in the work of Sawaya and Grossmann (2007) two issues may arise. Firstly, the continuous relaxation of LBM is often weak, leading to a high number of nodes enumerated in the branch and bound procedure. Secondly, the increase in the size of LCH due to the disaggregated variables and new constraints may not compensate the strengthening obtained in the relaxation, resulting in a high computational effort. In order to overcome these issues, Sawaya and Grossmann (2007) proposed a cutting plane methodology that consists in the generation of cutting planes obtained from the LCH and used to strengthen the relaxation of LBM. It is important to note, however, that in the last few years, MIP solvers have improved significantly in the use of the problem structure to reduce automatically the size of the formulation. As a result the emphasis should be placed on the strength of the relaxations rather than on the size of formulations. With this in mind, we present next the last developments in Linear GDPs.

Sawaya & Grossmann (2008) proved that any Linear Generalized Disjunctive Program (LGDP) that involves Boolean and continuous variables can be equivalently formulated as a Disjunctive Program (DP), that only involves continuous variables. This means that we are able to exploit the wealth of theory behind DP from Balas (1979,1985) in order to solve LGDP more efficiently.

One of the properties of disjunctive sets is that they can be expressed in many different equivalent forms. Among these forms, two extreme ones are the Conjunctive Normal Form (CNF), which is expressed as the intersection of elementary sets (i.e. sets that are the union of half spaces), and the Disjunctive Normal Form (DNF), which is expressed as the union of polyhedra. One important result in Disjunctive Programming Theory, as presented in the work of Balas (1985), is that we can systematically generate a set of equivalent DP formulations going from the CNF to the DNF by using an operation called **basic step** (Theorem 2.1, Balas (1985)), which preserves regularity. A basic step is defined as follows. Let F be the disjunctive set in RF given by $F = \bigcap_{j\in T} S_j$ where$S_j = \bigcup_{i\in Q_j} P_i$, Pi a polyhedron, i∈ Qj . For $k, l \in T$, $k \neq l$, a basic step consists in replacing $S_k \bigcap S_l$ with $S_{kl} = \bigcup_{\substack{i \in Q_k \\ j \in Q_l}} (P_i \bigcap P_j)$. Note that a basic step involves intersecting a given pair of disjunctions S_k and S_l.

Although the formulations obtained after the application of basic steps on the disjunctive sets are equivalent, their *continuous relaxations* are not. We denote the continuous relaxation of a disjunctive set $F = \bigcap_{j\in T} S_j$ in regular form where each *Sj* is a union of polyhedra, as the *hull-relaxation* of F (or *h-rel* F). Here $h-rel\, F := \bigcap_{j\in T} cl\, conv\, S_j$ and $cl\, conv\, S_j$ denotes the closure of the convex hull of *Sj*. That is, if $S_j = \bigcup_{i\in Q_j} P_i$, $P_i = \{x \in R^n, A^i x \leq b^i\}$, then $cl\, conv\, S_j$ is given by, $x = \sum_{i\in Q_j} v^i$, $\lambda_i \geq 0$, $\sum_{i\in Q_j} \lambda_i = 1$, $A^i v^i \leq b^i \lambda_i$ $i \in Q_j$. Note that the convex hull of F is in general different from its hull-relaxation.

As described by Balas (Theorem 4.3., Balas (1985)), the application of a basic step on a disjunctive set leads to a new disjunctive set whose relaxation is at least as tight, if not tighter, as the former. That is, for i=0,1,. . .,t let $F_i = \bigcap_{j\in T_i} S_j$ be a sequence of regular forms of a disjunctive set, such that: i) F0 is in CNF, with $P_0 = \bigcap_{j\in T_0} S_j$, ii) Ft is in DNF , iii) for i=1,. . .,t, Fi is obtained from Fi-1 by a basic step. Then $h-rel\, F_0 \supseteq h-rel\, F_1 \supseteq .... \supseteq h-rel\, F_t$. As shown by Sawaya and Grossmann (2008), this leads to a procedure to find MIP reformulations that are often tighter than the traditional LCH.

3.4.1. Illustration . Let us consider the following example:

$$
\begin{aligned}
&Min\, Z = x_2\\
&s.t.\ 0.5x_1 + x_2 \leq 1\\
&\begin{bmatrix} Y_1 \\ x_1 = 0 \\ x_2 = 0 \end{bmatrix} \vee \begin{bmatrix} \neg Y_1 \\ x_1 = 1 \\ 0 \leq x_2 \leq 1 \end{bmatrix} \qquad \text{(LGDP1)}\\
&0 \leq x_{1,2} \leq 1\\
&x_{1,2} \in R,\ Y_1 \in \{True, False\}
\end{aligned}
$$

An equivalent formulation can be obtained by the application of a basic step between the global constraint (or one term disjunction) $0.5x_1 + x_2 \leq 1$ and the two terms disjunction.

$$
\begin{aligned}
&Min\, Z = x_2\\
&s.t.\\
&\begin{bmatrix} Y_1 \\ x_1 = 0 \\ x_2 = 0 \\ 0.5x_1 + x_2 \leq 1 \end{bmatrix} \vee \begin{bmatrix} \neg Y_1 \\ x_1 = 1 \\ 0 \leq x_2 \leq 1 \\ 0.5x_1 + x_2 \leq 1 \end{bmatrix} \qquad \text{(LGDP2)}\\
&0 \leq x_{1,2} \leq 1\\
&x_{1,2} \in R,\ Y_1 \in \{True, False\}
\end{aligned}
$$

As it can be seen in the Fig. 4, the hull relaxation of the later formulation is tighter than the original leading to a stronger lower bound.

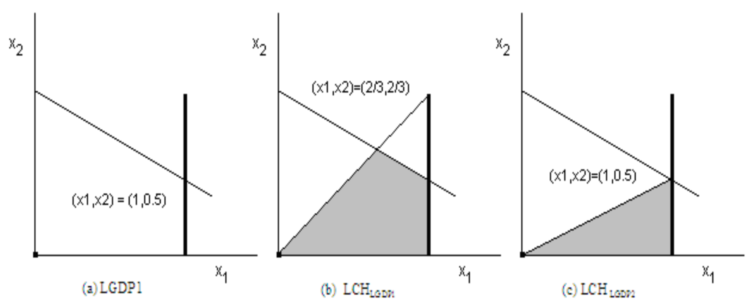

Fig. 4. *a-Projected feasible region of LGDP1 , b-Projected feasible region of relaxed LGDP1 , c-Projected feasible region of relaxed LGDP2*

3.4.2. Example. Strip Packing Problem (Hifi, 1998) . We apply the new approach to obtain stronger relaxations on a set of instances for the Strip Packing Problem. Given a set of small rectangles with width Hi and length Li and a large rectangular strip of fixed width W and unknown length L. The problem is to fit the small rectangules on the strip (without rotation and overlap) in order to minimize the length L of the strip

The LGDP for this problem is presented below (Sawaya & Grossmann, 2006).

$Min\, Z = lt$
$s.t.$
$lt \geq x_i + L_i \forall i \in N$

$$\begin{bmatrix} Y_{ij}^1 \\ x_i + L_i \leq x_j \end{bmatrix} \vee \begin{bmatrix} Y_{ij}^1 \\ x_i + L_i \leq x_j \end{bmatrix} \vee \begin{bmatrix} Y_{ij}^1 \\ x_i + L_i \leq x_j \end{bmatrix} \vee \begin{bmatrix} Y_{ij}^1 \\ x_i + L_i \leq x_j \end{bmatrix}$$

$x_i \leq UB_i - L_i \;\; \forall i \in N$

$H_i \leq y_i \leq W \;\; \forall i \in N$

$lt, x_i, y_i \in R_+^1,\, Y_{ij}^{1,2,3,4} \in \{True, False\} \forall i, j \in N, i < j$

In Table 2, the approach using basic steps to obtain stronger relaxations is compared with the original formulation.

Table 2: Comparison of sizes and lower bounds between original and new MIP reformulations

	Convex Hull Formulation				**Fromulation w. Basic Steps**			
Instance	**Vars**	**0-1**	**Constr.**	**LB**	**Vars**	**0-1**	**Constr.**	**LB**
4 Rectang.	102	24	143	4	170	24	347	8
25 Rectang.	4940	1112	7526	9	5783	1112	8232	27
31 Rectang.	9716	2256	14911	10.64	11452	2256	15624	33

It is important to note that although the size of the reformulated MIP is significantly increased when applying basic steps, the LB is greatly improved.

3.5. Nonconvex Generalized Disjunctive Programs. In general, some of the functions f , r_{ik} or g might be nonconvex, giving rise to a nonconvex GDP problem. The direct application of traditional algorithms to solve the reformulated MINLP in this case, such as Generalized Benders Decomposition (GBD) (Benders, 1962 and Geoffrion, 1972) or Outer Approximation (AO) (Viswanathan & Grossmann, 1990) may fail to find the global optimum since the solution of the NLP subproblem may correspond to a local optimum and the cuts in the master problem may not be valid. Therefore, specialized algorithms should be used in order to find the global optimum (Horst & Tuy, 1996 and Tawarmalani & Sahinidis, 2002).

With this aim in mind, Lee & Grossmann (2003) proposed the following two-level branch and bound algorithm.

The first step in this approach is to introduce convex underestimators of the nonconvex functions in the original nonconvex GDP. This leads to:

$$
\begin{aligned}
& Min\ Z = \bar{f}(x) + \textstyle\sum_{i\in D_k}\sum_{k\in K}\gamma_{ij}y_{ik} \\
& s.t.\ \bar{g}(x) \le 0 \\
& \bigvee_{i\in D_k}\begin{bmatrix} Y_{ik} \\ \bar{r}_{ik}(x)\le 0 \\ c_k = \gamma_{ik}\end{bmatrix} \qquad k\in K \qquad (\text{ RGDPNC }) \\
& \Omega(Y) = True \\
& x^{lo} \le x \le x^{up} \\
& x\in R^n\ ,\ c_k\in R^1,\ Y_{ik}\in\{True, False\}
\end{aligned}
$$

where $\bar{f}$, $\bar{r}_{ik}$, $\bar{g}$ are convex and the following inequalities are satisfied $\bar{f}(x)\le f(x)$, $\bar{r}_{ik}(x)\le r_{ik}(x)$, $\bar{g}(x)\le g(x)$. Note that suitable convex underestimators for these functions can be found in Tawarmalani & Sahinidis (2002)

The feasible region of *(RGDPNC)* can be relaxed by replacing each disjunction by its convex hull. This relaxation yields the following convex NLP

$$
\begin{aligned}
& Min\, Z = \bar{f}(x) + \textstyle\sum_{i\in D_k}\sum_{k\in K}\gamma_{ij}y_{ik} \\
& s.t.\ x = \textstyle\sum_{i\in D_K} v^{ik} \qquad k\in K \\
& \bar{g}(x)\le 0 \\
& y_{ik}\bar{r}_{ik}(\nu^{ik}/y_{ik}) \le 0 \qquad k\in K,\ i\in D_k \qquad (\text{RGDPRNC}) \\
& 0\le \nu^{ik} \le y_{ik}U_v \quad k\in K,\ i\in D_k \\
& \textstyle\sum_{i\in D_k} y_{ik} = 1 \quad k\in K \\
& Ay\ge a \\
& x\in R^n,\ v_{ik}\in R^1, c_k\in R^1, y_{ik}\in[0,1]\ \ k\in K,\ i\in D_k
\end{aligned}
$$

As proven in Lee & Grossmann (2003) the solution of this NLP formulation leads to a lower bound of the global optimum.

The second step consists in using the above relaxation to predict lower bounds within a spatial branch and bound framework. The main steps in this implementation are described in Fig. 5. The algorithm starts by obtaining a local solution of the nonconvex GDP problem by solving the MINLP reformulation with a local optimizer (e.g. DICOPT), which will give an upper bound of the solution (Z^U). Then, a bound contraction procedure is performed as described by Zamora and Grossmann (1999). Finally, a partial branch and bound method is used on $RGDP_{NC}$ as described in Lee & Grossmann (2003) that consists in only branching on the Boolean variables until a node with all the Boolean variables fixed is reached. At this point a spatial branch and bound procedure is performed as described in Quesada and Grossmann (1995).

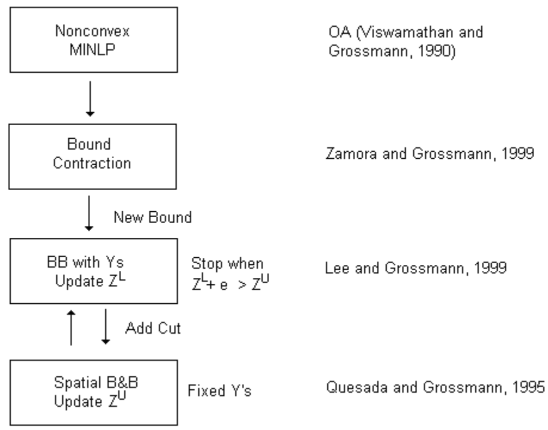

Fig. 5. *Steps in Global Optimization Algorithm*

While the method proved to be effective in solving several problems, a major question is whether one might be able to obtain stronger lower bounds to improve the computational efficiency.

Recently, Ruiz & Grossmann (2009) proposed an enhanced methodology that builds on the work of Sawaya & Grossmann (2008) to obtain stronger relaxations. The basic idea consists in relaxing the nonconvex terms in the GDP using valid linear over, underestimators previous to the application of basic steps. This leads to a new Linear GDP whose continuous relaxation is tighter and valid for the original nonconvex GDP problem. The implementation of basic steps is not trivial, Ruiz & Grossmann (2009) proposed a set of rules that aim at keeping the formulation small while improving the relaxation. Among others, it was shown that intersecting the global constraints with the disjunctions lead to a Linear GDP with the same number of disjuncts but a stronger relaxation.

The following example illustrates the idea behind this approach to obtain a stronger relaxation in a simple nonconvex GDP. Fig. 6 shows a small superstructure consisting of two reactors, each characterized by a flow-conversion curve, a conversion range for which it can be designed, and its corresponding cost as can be seen in Table 3. The problem consists in choosing the reactor and conversion that maximize the profit from sales of the product considering that there is a limit on the demand.

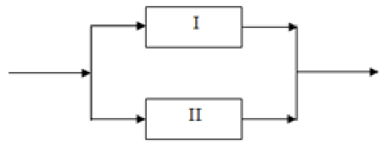

FIG. 6. *Two reactor network*

Table 3: Data for the reactors

Reactor	**Curve***		**Range**		**Cost**
	a	b	X^{lo}	X^{up}	Cp
I	-8	9	0.2	0.95	2.5
II	-10	15	0.7	0.99	1.5

The characteristic curve is defined as F = aX + b in the range of conversions $[X^{lo}, X^{up}]$ where F and X are the flow of raw material and conversion respectively.

The bilinear GDP model, which maximizes the profit, can be stated as follows:

$$Max\, Z = \theta FX - \gamma F - CP$$

$$s.t. \quad FX \leq d$$

$$\begin{bmatrix} Y_{11} \\ F = \alpha_1 X + \beta_1 \\ X_1^{lo} \leq X \leq X_1^{up} \\ CP = Cp_1 \end{bmatrix} \vee \begin{bmatrix} Y_{21} \\ F = \alpha_2 X + \beta_2 \\ X_2^{lo} \leq X \leq X_2^{up} \\ CP = Cp_2 \end{bmatrix} \quad (GDP1_{NC})$$

$$Y_{11} \underline{\vee} Y_{11} = True$$

$$X, F, CP \in R^1, F^{lo} \leq F \leq F^{up}, Y_{11}, Y_{21} \in \{True, False\}$$

The associated Linear GDP relaxation is obtained by replacing the bilinear term, *FX*, using the McCormick convex envelopes:

$$Max\, Z = \theta P - \gamma F - CP$$

$$s.t. \quad P \leq d$$

$$P \leq FX^{lo} + F^{up}X - F^{up}X^{lo};\ P \leq FX^{up} + F^{lo}X - F^{lo}X^{up}$$

$$P \geq FX^{lo} + F^{lo}X - F^{lo}X^{lo};\ P \leq FX^{up} + F^{up}X - F^{up}X^{up}$$

$$\begin{bmatrix} Y_{11} \\ F = \alpha_1 X + \beta_1 \\ X_1^{lo} \leq X \leq X_1^{up} \\ CP = Cp_1 \end{bmatrix} \vee \begin{bmatrix} Y_{21} \\ F = \alpha_2 X + \beta_2 \\ X_2^{lo} \leq X \leq X_2^{up} \\ CP = Cp_2 \end{bmatrix} \quad (GDP1_{RLP0})$$

$Y_{11} \underline{\vee} Y_{11} = True$

$X, F, CP \in R^1, F^{lo} \leq F \leq F^{up}, Y_{11}, Y_{21} \in \{True, False\}$

Intersecting the improper disjunctions given by the inequalities of the relaxed bilinear term with the only proper disjunction (i.e. by applying five basic steps), we obtain the following GDP formulation,

$$Max\, Z = \theta P - \gamma F - CP \qquad (GDP1_{RLP1})$$

$s.t.$

$$\begin{bmatrix} Y_{11} \\ P \leq d \\ P \leq FX^{up} + F^{lo}X - F^{lo}X^{up} \\ P \leq FX^{lo} + F^{up}X - F^{up}X^{lo} \\ P \geq FX^{lo} + F^{lo}X - F^{lo}X^{lo} \\ P \geq FX^{up} + F^{up}X - F^{up}X^{up} \\ F = \alpha_1 X + \beta_1 \\ X_1^{lo} \leq X \leq X_1^{up} \\ CP = Cp_1 \end{bmatrix} \vee \begin{bmatrix} Y_{21} \\ P \leq d \\ P \leq FX^{up} + F^{lo}X - F^{lo}X^{up} \\ P \leq FX^{lo} + F^{up}X - F^{up}X^{lo} \\ P \geq FX^{lo} + F^{lo}X - F^{lo}X^{lo} \\ P \geq FX^{up} + F^{up}X - F^{up}X^{up} \\ F = \alpha_2 X + \beta_2 \\ X_2^{lo} \leq X \leq X_2^{up} \\ CP = Cp_2 \end{bmatrix}$$

$Y_{11} \underline{\vee} Y_{11} = True$

$X, F, CP \in R^1, F^{lo} \leq F \leq F^{up}, Y_{11}, Y_{21} \in \{True, False\}$

Fig. 7 shows the actual feasible region of $(GDP1_{NC})$ and the projection on the F-X space of the hull relaxations of $(GDP1_{RLP0})$ and $(GDP1_{RLP1})$, where clearly the feasible space in $(GDP1_{RLP1})$ is tighter than in $(GDP1_{RLP0})$. Notice that in this case the choice of reactor II is infeasible.

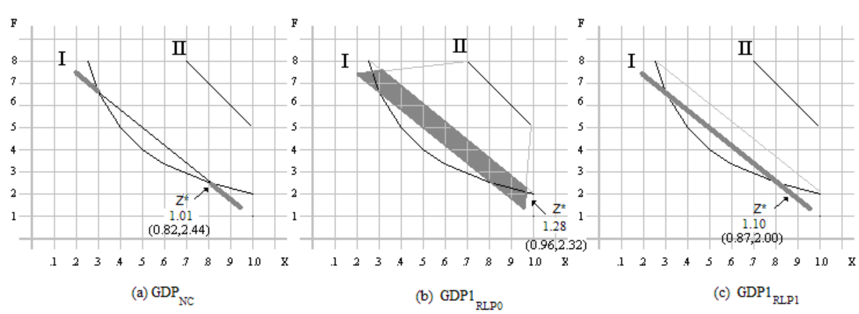

FIG. 7. *a-Projected feasible region of* $GDP1_{NC}$ *, b-Projected feasible region of relaxed* $GDP1_{RLP0}$*, c-Projected feasible region of relaxed* $GDP1_{RLP1}$

3.5.1. Example. Water Treatment Network (Galan and Grossmann , 1998). This example corresponds to a synthesis problem of a distributed wastewater multicomponent network (See Fig 8), which is taken from Galan and Grossmann (1998). Given a set of process liquid streams with known composition, a set of technologies for the removal of pollutants, and a set of mixers and splitters, the objective is to find the interconnections of the technologies and their flowrates to meet the specified discharge composition of pollutant at minimum total cost. Discrete choices involve deciding what equipment to use for each treatment unit.

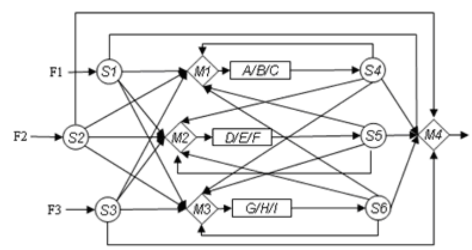

FIG. 8. *Water treatment superstructure*

Lee and Grossmann (2003) formulated this problem as the following nonconvex GDP problem:

$$Min\, Z = \sum_{k \in PU} CP_k$$

$$f_k^j = \sum_{i \in M_k} f_i^j \qquad \forall jk \in MU$$

$$\sum_{i \in S_k} f_i^j = f_k^j \qquad \forall jk \in SU$$

$$\sum_{i \in S_k} \zeta_i^k = 1 \qquad k \in SU$$

$$f_i^j = \zeta_i^k f_k^j \quad \forall j \quad i \in S_k \quad k \in SU$$

$$\bigvee_{h \in D_k} \begin{bmatrix} YP_k^h \\ f_i^j = \beta_k^{jh} f_{i'}^j, i \in OPU_k, i' \in IPU_k, \forall j \\ F_k = \sum_j f_i^j, i \in OPU_k \\ CP_k = \partial_{ik} F_k \end{bmatrix} \qquad k \in PU$$

$$0 \le \zeta_i^k \le 1 \quad \forall j, k$$
$$0 \le f_i^j, f_k^j \quad \forall i, j, k$$
$$0 \le CP_k \quad \forall k$$
$$YP_k^h \in \{true, false\} \ \forall h \in D_k \ \forall k \in PU$$

The problem involves 9 discrete variables and 114 continuous variables with 36 bilinear terms. Table 4 shows the computational performance when the Lee and Grossmann (2003) relaxation is used within a spatial branch and bound framework with the one proposed in Ruiz & Grossmann (2009) work.

Table 4 Lower bounds of proposed framework

Global Optimum	Lower Bound (Lee & Grossmann Relaxation)	Lower Bound (Ruiz & Grossmann Relaxation)	Best Lower Bound
1214.87	400.66	431.9	431.9

As it can be seen, an improved lower bound was obtained (i.e. 431.9 vs 400.66) which is a direct indication of the reduction of the relaxed feasible region. The column "Best Lower Bound", can be used as an indicator of the performance of the proposed set of rules to apply basic steps. Note that the lower bound obtained in this new approach is the same as the one obtained by solving the relaxed DNF, which is quite remarkable. A further indication of tightening is shown in Table 5 where numerical results of the branch and bound algorithm proposed in section 6 are presented. As it can be seen the number of nodes that the spatial branch and bound algorithm requires before finding the global solution is significantly reduced.

Table 5 Performance of proposed methodology with spatial B&B.

	Global Optimization Technique using Lee & Grossmann Relaxation			Global Optimization Technique using Ruiz & Grossmann Relaxation		
Global Optimum	Nodes	Bound contract. (% Avg)	CPU Time (sec)	Nodes	Bound contract. (% Avg)	CPU Time (sec)
1214.87	408	8	176	130	16	115

Table 6 shows the size of the LP relaxation obtained in each methodology. Note that although the proposed methodology leads to a significant increase in the size of the formulation, this is not translated proportionally to the solution time of the resulting LP. This behavior can be understood by considering that in general, the LP pre-solver will take advantage of the particular structures of these LPs.

Table 6 Size of the LP relaxation for example problems

Size of the LP Relaxation			
Lee & Grossmann		Ruiz & Grossmann	
Constraints	Variables	Constraints	Variables
544	346	3424	1210

Aknowledgments

The authors would like to acknowledge financial support from the National Science Foundation under Grant OCI-0750826.

REFERENCES

[1] Abhishek,K., S. Leyffer, and J. T. Linderoth, *FilMINT: An Outer-Approximation-Based Solver for Nonlinear Mixed Integer Programs*, ANL/MCS-P1374-0906,Argonne National Laboratory,2006

[2] Balas, E. , *Disjunctive Programming* , vol 5 , 1979 , pp 3-51

[3] Balas, E. , *Disjunctive Programming and a hierarchy of relaxations for discrete optimization problems*, vol 6 ,1985, SIAM J. Alg. Disc. Meth., pp 466-486

[4] Beaumont, N.,*An Algorithm for Disjunctive Programs*,vol 48, 1991, European Journal of Operations Research, pp 362-371

[5] Belotti, P.,*Couenne User's Manual*,2008, https://projects.coin-or.org/Couenne/browser/trunk/Couenne/doc/couenne-user-manual.pdf

[6] Benders J. F.,*Partitioning procedures for solving mixed-variables programming problems*,vol 4, 1962, Numer. Math., pp 238-252

[7] Borchers, B. and J.E. Mitchell,*An Improved Branch and Bound Algorithm for Mixed Integer Nonlinear Programming*, vol 21, 1994, Computers and Operations Research, pp 359-367

[8] Bonami, P., L.T. Biegler, A.R. Conn, G. Cornuejols, I.E. Grossmann, C.D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, A. Wchter,*An algorithmic framework for convex mixed integer nonlinear programs*, vol 5, 2008, Discrete Optimization, pp 186-204

[9] Brooke A., Kendrick D., Meeraus A., Raman R,*GAMS, a User's Guide*,1998,GAMS Development Corporation, Washington

[10] Duran, M.A. and I.E. Grossmann,*An Outer-Approximation Algorithm for a Class of Mixed-integer Nonlinear Programs*,vol 36,1986, Math Programming, pp 307

[11] Fletcher, R. and S. Leyffer,*Solving Mixed Integer Nonlinear Programs by Outer Approximation*,vol 66,1994, Math Programming, pp 327

[12] Galan, B. Grossmann, I. E.,*Optimal Design of Distributed Wastewater Treatment Networks*, vol 37, 1998, Ind. Eng. Chem. Res., pp 4036-4048

[13] Geoffrion A. M., *Generalized Benders decomposition*, vol 10, 1972, JOTA, pp 237-260

[14] Grossmann, I. E. , *Review of Non-Linear Mixed Integer and Disjunctive Programming Techiques for Process Systems Engineering*, vol 3, 2002,Optimization and Engineering, pp 227-252

[15] Grossmann, I.E., J.A. Caballero and H. Yeomans,*Advances in Mathematical Programming for Automated Design, Integration and Operation of Chemical Processes*,vol 16,1999, Korean J. Chem. Eng., pp 407-426

[16] Grossmann, I.E. and Lee, S., *Generalized Convex Disjunctive Programming: Nonlinear Convex Hull Relaxation*, vol 26, 2003, Computational Optimization and Applications, pp 83-100

[17] Gupta, O. K. and Ravindran, V.,*Branch and Bound Experiments in Convex Nonlinear Integer Programming*,vol 31(12), 1985, Management Science, pp 1533-1546

[18] Hooker, J.N. and M.A. Osorio,*Mixed logical.linear programming*, vol 96-97 ,1999, Discrete Applied Mathematics, pp 395-442

[19] Hooker, J.N., *Logic-Based Methods for Optimization: Combining Optimization and Constraint Satisfaction*,2000, Wiley

[20] Horst, R. Tuy, H.,*Global Optimization deterministic approaches*,3rd Ed, Springer-Verlag, 1996

[21] Kallrath, J.,*Mixed Integer Optimization in the Chemical Process Industry: Experience, Potential and Future*, vol 78,2000, Trans. I .Chem E., pp 809-822

[22] Lee, S. and I.E. Grossmann, *New Algorithms for Nonlinear Generalized Disjunctive Programming*,vol 24, 2000, Computers and Chemical Engineering, pp 2125-2141

[23] Lee, S. Grossmann I. E., *Global optimization of nonlinear generalized disjunc-*

tive programming with bilinear inequality constraints: application to process networks,vol 27, 2003, Computers and Chemical Engineering, pp 1557-1575

[24] Leyffer, S., *Integrating SQP and Branch and Bound for Mixed Integer Nonlinear Programming*,vol 18, 2001, Computational Optimization and Applications, pp 295-309

[25] Lindo Systems Inc, *LindoGLOBAL Solver*

[26] Mndez, C.A., Cerd, J., Grossmann, I.E., Harjunkoski, I., Fahl, M.,*State-of-the-art Review of Optimization Methods for Short-Term Scheduling of Batch Processes*,vol 30, 2006, Comput. Chem. Eng., pp 913.

[27] Nemhauser, G. L. and Wolsey, L. A.,*Integer and Combinatorial Optimization*, 1988, Wiley-Interscience

[28] Quesada, I. and I.E. Grossmann, *An LP/NLP Based Branch and Bound Algorithm for Convex MINLP Optimization Problems*,vol 16, 1992, Computers and Chemical Engineering, pp 937-947

[29] Quesada, I. Grossmann I. E.,*Global optimization of bilinear process networks with multicomponent flows*,vol 19(12), 1995, Computers and Chemical Engineering, pp 1219-1242

[30] Raman, R. and I.E. Grossmann,*Relation Between MILP Modelling and Logical Inference for Chemical Process Synthesis*,vol 15, 1991, Computers and Chemical Engineering, pp 73

[31] Raman, R. and I.E. Grossmann, *Modelling and Computational Techniques for Logic Based Integer Programming*,vol 18, 1994, Computers and Chemical Engineering, pp 563

[32] Ruiz, J. P. and Grossmann I. E. , *Strengthening the lower bounds for bilinear and concave GDP problems*, Submitted for publication (2009)

[33] Sahinidis, N. V.,*BARON: A General Purpose Global Optimization Software Package*,vol 8(2),1996, Journal of Global Optimization,201-205

[34] Sawaya and Grossmann,*Thesis: Reformulations, relaxations and cutting planes for generalized disjunctive programming*,2006, Carnegie Mellon University

[35] Sawaya N., Grossmann I. E. and Furman K., *Computational Implementation of Non-Linear Convex Hull Reformulation*. Submitted for publication (2007)

[36] Sawaya N. and Grossmann I. E.,*Reformulations, Relaxations and Cutting Planes for Linear Generalized Disjunctive Programming*. Submitted for publication (2008)

[37] Schweiger C.A. and C.A. Floudas,*Process Synthesis, Design and Control: A Mixed Integer Optimal Control Framework*, 1998, Proceedings of DYCOPS-5 on Dynamics and Control of Process Systems, pp. 189-194

[38] Stubbs R. and S. Mehrotra, *A Branch-and-Cut Method for 0-1 Mixed Convex Programming*, vol 86(3), 1999, Mathematical Programming, pp 515-532

[39] Tawarmalani, M. and Sahinidis, N. ,*Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming*, 2002 , Kluwer Academic Publishers

[40] Trkay, M. and I.E. Grossmann,*A Logic Based Outer-Approximation Algorithm for MINLP Optimization of Process Flowsheets*,vol 20,1996, Computers and Chemical Enginering, 959-978

[41] Vecchietti, A., Lee, S. , Grossmann, I. E.,*Modeling of discrete/continuous optimization problems: characterization and formulation of disjunctions and their relaxations*, vol 27, 2003, Computers and Chemical Engineering, pp 433-448

[42] Vecchietti, A. and Grossmann, I.E.,*LOGMIP: A Discrete Continuous Nonlinear Optimizer*,vol 23, 2003, Computers and Chemical Engineering, pp 555-565

[43] Viswanathan Grossmann I.E.,*A combined penalty function and outer-approximation method for MINLP optimization*,vol 14,1990, Computers and Chemical Engineering, pp 769-782

[44] Westerlund, T. and F. Pettersson, *A Cutting Plane Method for Solving Convex MINLP Problems*,vol 19, 1995, Computers and Chemical Engineering,

S131-S136

[45] Westerlund T. and Porn R.,*Solving Pseudo-Convex Mixed Integer Optimization Problems by Cutting Plane Techniques*, vol 3, 2002, Optimization and Engineering, 3, 253-280.

[46] Williams, H.P., *Mathematical Building in Mathematical Programming*,1985, John Wiley

[47] Yuan, X., S. Zhang, L. Piboleau and S. Domenech, *Une Methode d'optimisation Nonlineare en Variables Mixtes pour la Conception de Procedes*,vol 22, 1988, RAIRO, pp 331

[48] Zamora, J. M. , Grossmann, I. E.,*A branch and bound algorithm for problems with concave univariate , bilinear and linear fractional terms*, vol 14(3),1999, Journal of Global Optimization, 217-249