2-2014

Review of Mixed-Integer Nonlinear and Generalized Disjunctive Programming Methods

Francisco Trespalacios
Carnegie Mellon University

Ignacio E. Grossmann
Carnegie Mellon University, grossmann@cmu.edu

Published In
Chemie Ingenieur Technik, 86, 7, 991-1012.

Review of mixed-integer nonlinear and generalized disjunctive programming methods in Process Systems Engineering

Ignacio E. Grossmann* and Francisco Trespalacios
Department of Chemical Engineering
Carnegie Mellon University, Pittsburgh, PA

February, 2014

Abstract

This work presents a review of the applications of mixed-integer nonlinear programming (MINLP) in process systems engineering (PSE). A review on the main deterministic MINLP solution methods is presented, including an overview of the main MINLP solvers. Generalized disjunctive programming (GDP) is an alternative higher-level representation of MINLP problems. This work reviews some methods for solving GDP models, and techniques for improving MINLP methods through GDP. The paper also provides a high-level review of the applications of MINLP in PSE, particularly in process synthesis, planning and scheduling, process control and molecular computing.

1 Introduction

Many optimization problems require the modeling of discrete and continuous variables, giving rise to mixed-integer linear and mixed-integer nonlinear programming (MILP/MINLP). Models in which the objective function and the constraints are linear are MILP problems. MINLP problems involve nonlinear objective function and/or nonlinear constraints. MINLP is divided in two categories, convex MINLP and nonconvex MINLP. Convex MINLP involves minimizing a convex objective function over a convex feasible region. In nonconvex MINLP the objective function and/or the feasible region are not convex. Many Process Systems Engineering (PSE) applications are modelled using MINLP. Process synthesis[1], planning and scheduling[2], process control[3], and molecular computing[4] are some of the applications of MINLP in PSE.

MILP theory[5–8] has been considerably enriched in the past decades, which has been reflected in the advances of methods to solve this type of problems[9]. Some of the powerful commercial solvers are CPLEX[10], Gurobi[11], and XPRESS-MP[12]. A survey the noncommercial solvers is given by Linderoth and Ralphs[13].

MINLP methods greatly benefit from advances in MILP and NLP methods; however, the progress have been more modest. Belotti et al.[14] provide a comprehensive review of mixed-integer nonlinear optimization. Bonami et al.[15] review convex MINLP algorithms and software in more detail. Tawarmalani and Sahinidis[16] describe global optimization theory, algorithms and applications.

An alternative representation of MINLP is Generalized Disjunctive Programming (GDP)[17]. GDP models are represented through continuous and Boolean variables, and involve algebraic equations, disjunctions and logic propositions. GDP models are typically reformulated as MINLP problems to exploit the developments in these solvers[7, 18, 19]. However, special techniques can help to improve the performance in solving GDPs.

The purpose of this work is to present the main MINLP and GDP deterministic methods and applications in PSE. This paper is organized as follows. Section 2 provides an overview of MINLP and GDP, and relevant concepts that are recurrent in the solving techniques. Section 3 presents the main algorithms and solvers for convex MINLPs. Section 4 presents specialized algorithms for solving convex GDPs. It also shows how, in some cases, these algorithms perform better than solving an MINLP reformulation of the GDP. The traditional deterministic method for global optimization is presented in section 5. Section 6 presents GDP techniques that can help to improve the performance of global solvers. It also shows some particular applications for PSE in which GDP techniques help to obtain better or faster solutions. Section 7 provides a summary of some of the applications of MINLP and GDP in PSE. Finally, we provide some concluding remarks in section 8.

2 Background

In this section we provide some basic definitions for MINLP and for GDP. In the MINLP background we present the MINLP general form, and three NLP problems that are related and relevant to the original MINLP. In the GDP background, we first present the GDP general form and two alternative MINLP reformulations. We then present three NLP problems, that are related and relevant to the original GDP.

2.1 MINLP basic definitions

The general form of an MINLP model is as follows:

$$\begin{aligned} & & min\ z = f(x,y) \\ s.t. & & g(x,y) \leq 0 \\ & & x \in X \\ & & y \in Y \end{aligned} \qquad \text{(MINLP)}$$

where $f(\cdot)$ and $g(\cdot)$ are twice differentiable functions, x are the continuous variables and y the discrete ones. For practical applications, the set X is assumed to be a convex compact set ($X = \{x|\ x \in \mathbb{R}^n,\ Dx \leq d,\ x^{lo} \leq x \leq x^{up}\}$). The polyhedral set of integer points is represented by the set Y ($Y = \{y|\ y \in \mathbb{Z}^m,\ Ay \leq a\}$). Note that this form allows for more general constraints such as equalities, linear constraints and lower and upper bounds. It can also be used to maximize a given function ($maxf(\cdot) = -min - f(\cdot)$).

It is important to note than in most applications of interest the integer variables are restricted to 0-1 values ($y \in \{0,1\}^m$)). Also, the constraint functions $f(\cdot)$, $g(\cdot)$ are typically linear in y ($f(x,y) = r(x) + c^T y$, $g(x,y) = h(x) + By$)[20].

There are three NLP subproblems that arise from (MINLP), and that are commonly used in the solving techniques. The first one is the *continuous relaxation* of (MINLP):

$$\begin{aligned} & min\ z = f(x,y) \\ s.t. \quad & g(x,y) \leq 0 \\ & x \in X \\ & y \in Y_R \end{aligned} \tag{r-MINLP}$$

where Y_R is the continuous relaxation of Y ($Y = \{y|\ y \in \mathbb{R}^m,\ Ay \leq a\}$). Note that in (r-MINLP) the integer variables y are treated as continuous.

The second relevant NLP is the *NLP subproblem for a fixed* y^p:

$$\begin{aligned} & min\ z = f(x,y^p) \\ s.t. \quad & g(x,y^p) \leq 0 \\ & x \in \mathbb{R}^n \end{aligned} \tag{fx-MINLP}$$

It is important to note that (fx-MINLP) provides an upper bound for (MINLP) when it has a feasible solution, while (r-MINLP) provides a lower bound. It is also important to note that a combination of these two extreme NLPs can be generated, by fixing some of the integer variables, while continuously relaxing the remaining ones.

When (fx-MINLP) is not feasible, the following *feasibility problem* is considered:

$$\begin{aligned} & min\ u \\ s.t. \quad & g_j(x,y^p) \leq u \quad j \in J \\ & x \in X \\ & u \in \mathbb{R} \end{aligned} \tag{feas-MINLP}$$

where J is the index set for inequalities. (feas-MINLP) seeks to minimize the infeasibility of the most violated constraint.

2.2 GDP basic definitions

Generalized Disjunctive Programming[17, 19, 21] is an alternative method to formulate optimization problems. It can be regarded as an extension of disjunctive programming[22].

The GDP general form is as follows:

$$
\begin{aligned}
& min\ z = f(x) \\
s.t.\quad & g(x) \leq 0 \\
& \bigvee_{i \in D_k} \begin{bmatrix} Y_{ki} \\ r_{ki}(x) \leq 0 \end{bmatrix} && k \in K \\
& \bigvee_{i \in D_k} Y_{ki} && k \in K \\
& \Omega(Y) = True \\
& x^{lo} \leq x \leq x^{up} \\
& x \in \mathbb{R}^n \\
& Y_{ki} \in \{True, False\} && k \in K, i \in D_k
\end{aligned}
\tag{GDP}
$$

(GDP) seeks to minimize a function of the continuous variables x. $g(x)$ are the global constraints that need to be satisfied independently of the discrete decisions. (GDP) contains $k \in K$ disjunctions. Each one involves $i \in D_k$ terms, linked together by an OR operator ($\vee$). Each disjunctive term is associated with a Boolean variable Y_{ki} and a set of inequalities $r_{ki}(x) \leq 0$. One and only one Boolean variable can be $True$ ($\bigvee_{i \in D_k} Y_{ki}$). When a disjunctive term is selected ($Y_{ki} = True$), the corresponding inequalities are enforced. Otherwise, the constraints are ignored. The logic relations among the Boolean variables are represented by $\Omega(Y) = True$.

In order to exploit the advances in MINLP solvers, GDP models can be reformulated as MINLP problems using the Big-M[7] (BM) or Hull Reformulation[18] (HR). The (BM) reformulation is given as follows:

$$
\begin{aligned}
& min\ z = f(x) \\
s.t.\quad & g(x) \leq 0 \\
& r_{ki}(x) \leq M^{ki}(1 - y_{ki}) && k \in K, i \in D_k \\
& \sum_{i \in D_k} y_{ki} = 1 && k \in K \\
& Hy \geq h \\
& x^{lo} \leq x \leq x^{up} \\
& x \in \mathbb{R}^n \\
& y_{ki} \in \{0, 1\} && k \in K, i \in D_k
\end{aligned}
\tag{BM}
$$

The (HR) formulation is given as follows:

$$
\begin{aligned}
& min\ z = f(x) \\
s.t.\quad & g(x) \leq 0 \\
& x = \sum_{i \in D_k} \nu^{ki} && k \in K \\
& y_{ki} r_{ki}(\nu^{ki}/y_{ki}) \leq 0 && k \in K, i \in D_k \\
& \sum_{i \in D_k} y_{ki} = 1 && k \in K \\
& Hy \geq h \\
& x^{lo} y_{ki} \leq \nu^{ki} \leq x^{up} y_{ki} && k \in K, i \in D_k \\
& x \in \mathbb{R}^n \\
& y_{ki} \in \{0,1\} && k \in K, i \in D_k
\end{aligned}
\tag{HR}
$$

In both MINLP reformulations, the Boolean variables Y_{ki} are transformed into 0-1 variables y_{ki}, $Y_{ki} = True$ is equivalent to $y_{ki} = 1$, while $Y_{ki} = False$ is equivalent to $y_{ki} = 0$. The logic relations ($\Omega(Y) = True$) are transformed into integer linear constraints ($Hy \geq h$). This transformation is well-known and simple to obtain[19, 23–25]. The equation $\sum_{i \in D_k} y_{ki} = 1$ guarantees that only one disjunctive term is selected per disjunction.

In the (BM) a M^{ki} parameter is introduced, so when a term is selected ($y_{ki} = 1$) the corresponding constraints $r_{ki}(x) \leq 0$ are enforced. When it is not selected ($y_{ki} = 0$), and M^{ki} is large enough, the corresponding constraint $r_{ki}(x) \leq M^{ki}$ becomes redundant.

In (HR), the continuous variables x are disaggregated into variables ν^{ki}, for each disjunctive term $i \in D_k$ in each disjunction $k \in K$. The constraint $x^{lo} y_{ki} \leq \nu^{ki} \leq x^{up} y_{ki}$ enforces that, when a term is active ($y_{ki} = 1$), the corresponding disaggregated variable lies within the variable bounds. When it is not selected it takes a value of zero. The constraint $x = \sum_{i \in D_k} \nu^{ki}$ enforces that the original variable x has the same value as the disaggregated variables of the active terms. The constraints of a disjunctive term are represented by the perspective function $y_{ki} r_{ki}(\nu^{ki}/y_{ki})$, so when a term is active the constraint is enforced. When it is not active, the constraint is trivially satisfied. When the constraints in the disjunction are linear ($A^{ki} x \leq a^{ki}$), the perspective function becomes $A^{ki} \nu^{ki} \leq a^{ki} y_{ki}$. To avoid singularities in the perspective function, the following approximation can be used[26]:

$$
y_{ki} r_{ki}(\nu^{ki}/y_{ki}) \approx ((1-\epsilon) y_{ki} + \epsilon) r_{ki}\left(\frac{\nu^{ki}}{(1-\epsilon) y_{ki} + \epsilon}\right) - \epsilon r_{ki}(0)(1 - y_{ki}) \tag{APP}
$$

where ϵ is a small finite number (e.g. 10^-5). It is important to note that this approximation yields an exact value at $y_{ki} = 0$ and $y_{ki} = 1$, and is convex if r_{ki} is convex.

The (BM) reformulation generates a smaller MINLP, while the (HR) provides a tighter formulation[27,

28].

In a similar manner to the MINLP, there are three relevant NLP problems. For the continuous relaxation of the GDP, it is necessary to use an MINLP reformulation. For the reformulations presented in this paper, (BM) and (HR), we denote their continuous relaxations as (r-BM) and (r-HR) respectively. The (r-HR) will always provide a lower bound that is as good, and usually better, than the (r-BM).

The second relevant NLP is the *NLP subproblem for a fixed alternative* KI^p:

$$\begin{aligned} & min\ z = f(x) \\ s.t. \quad & g(x) \leq 0 \\ & r_{ki}(x) \leq 0 \qquad ki \in KI^p \\ & x^{lo} \leq x \leq x^{up} \\ & x \in \mathbb{R}^n \end{aligned} \tag{fx-GDP}$$

If the problem is infeasible, the following *feasibility problem* is considered:

$$\begin{aligned} & min\ u \\ s.t. \quad & g_e(x) \leq u \qquad\qquad e \in E \\ & r_{kis}(x) \leq u \qquad kis \in KIS^p \\ & x^{lo} \leq x \leq x^{up} \\ & (x, u) \in \mathbb{R}^{n+1} \end{aligned} \tag{feas-GDP}$$

where E is the set of global constraints, and KIS^p is the set of constraints of the selected disjunctive terms KI^p.

3 Convex MINLP methods

In this section we first present convex MINLP and its MILP relaxation. We then describe the five common methods for solving convex MINLP. Afterwards, we illustrate the performance of the methods with a convex MINLP example. Finally, we describe some of the convex MINLP solvers and the methods they use.

3.1 Convex MINLP and its linear relaxation

A convex MINLP is a special case of MINLP in which (r-MINLP) is convex. In particular, if $f(\cdot)$ and $g(\cdot)$ are convex functions, then the problem is a convex MINLP.

The convexity of a nonlinear function can be exploited, in order to generate a linear approximation of the problem for a given set of points. This is done by generating a first-order Taylor series approximation of $f(\cdot)$ and $g(\cdot)$ at a given set of points $p = 1, ..., P$. This relaxation is represented as follows:

$$
\begin{aligned}
& min\ \alpha \\
s.t. \quad & f(x^p, y^p) + \nabla f(x^p, y^p)^T \begin{bmatrix} x - x^p \\ y - y^p \end{bmatrix} \leq \alpha \qquad p = 1, ..., P \\
& g_j(x^p, y^p) + \nabla g_j(x^p, y^p)^T \begin{bmatrix} x - x^p \\ y - y^p \end{bmatrix} \leq 0 \quad j \in J^p; p = 1, ..., P \\
& x \in X \\
& y \in Y
\end{aligned}
\tag{M-MIP}
$$

where $J^p \subseteq J$, since a subset of the linearizations can be included. Note that, due to convexity, (M-MIP) is a relaxation of (MINLP), therefore it provides a valid lower bound[20].

3.2 Nonlinear branch and bound

The nonlinear branch and bound[29–33] is an extension to the well known linear branch and bound[5]. To find optimality, the method performs an extensive tree search on the integer variables. It first solves the continuous relaxation of the MINLP. If the solution assigns integer values to all integer variables, then it is optimal and the algorithm stops. If it is not, an integer variable whose value at the current node is not integer is selected ($y_i = y_i^{(0)}$). A branching is performed in this variable, giving rise to two new NLP problems. One NLP includes the bound $y_i \leq \lfloor y_i^{(0)} \rfloor$, while the other one $y_i \geq \lceil y_i^{(0)} \rceil$. These procedure is repeated until the tree search is exhausted. If an integer feasible solution is found (i.e. the solution provides integer values to all the integer variables), then it provides an upper bound. There are two cases in which some of the nodes are pruned, which makes the branch and bound method faster than enumerating every node. The first case in which a node is pruned occurs when the NLP that corresponds to the node is infeasible. The second case occurs when the solution of the NLP of the node is larger than the current upper bound. The branch and bound algorithm is described as follows:

For a node N_p, let z^p denote the optimal value of the corresponding NLP_p, and (x^p, y^p) its solution. Let $\mathcal{L}$ be the set of nodes to be solved, and $NLP_{(0)}$ be (r-MINLP). Let z^{lo} and z^{up} be, respectively, a lower and upper bound of the optimal value of the objective function z^*.

0. Initialization. $\mathcal{L} = N_{(0)}$, $z^{up} = \infty$, $(x^*, y^*) = \emptyset$.

1. Terminate? If $\mathcal{L} = \emptyset$, then (x^*, y^*) is optimal.

2. Node selection. Choose a node N_p in $\mathcal{L}$, and delete it from $\mathcal{L}$.

3. Bound. Solve NLP_p. If it is infeasible, go to step 1. Else, let z^p be the objective value, and (x^p, y^p) its solution.

4. Prune. If $z^p \geq z^{up}$, go to step 1.

If $y^p \in \mathbb{Z}^m$, let $z^{up} = z^p$ and $(x^*, y^*) = (x^p, y^p)$. Delete all nodes N_q from $\mathcal{L}$ in which $z^q \geq z^{up}$. Go to step 1.

If $y^p \notin \mathbb{Z}^m$, go to step 5

5. Branch. Select a variable y_i, such that $y_i^p \notin \mathbb{Z}$. Construct NLP_p^1 and NLP_p^2, by using the same continuous relaxation of NLP_p, but adding one of the constraints $y_i \leq \lfloor y_i^p \rfloor$, and $y_i \geq \lceil y_i^p \rceil$, in each of the problems. Add the two new nodes (corresponding to NLP_p^1 and NLP_p^2) to $\mathcal{L}$, and go to step 1.

The performance of the algorithm strongly depends on the selection of branching variables and node selection strategies. Presolve and cutting planes can improve the performance of the traditional branch and bound algorithm. These topics are out of the scope of this paper, but we refer the reader to the recent work by Belotti et al.[14] and Bonami et al.[15] for a more detailed review of these concepts.

3.3 Outer approximation

Outer approximation[34, 35] makes use of two main problems: (M-MIP) and (fx-MINLP). The main idea is to use the approximate linear problem (M-MIP) to find a lower bound (z^{lo}) and obtain an integer solution to the approximate problem (y^p). This lower bounding problem is called master problem. For the subproblem, the binary variable y^p is fixed, and then (fx-MINLP) is solved. If the solution to (fx-MINLP) is feasible, then it provides an upper bound z^{up}. If it is not, (feas-MINLP) is solved to provide information about the subproblem, and an inequality that cuts off that integer solution is added. This method is performed iteratively until the gap of z^{up} and z^{lo} is less than the specified tolerance. At each iteration, the subproblem provides a solutions (x^p, y^p), which is then included in the master problem (M-MIP) to improve the approximation. Since the function linearizations are accumulated, the lower bounding problem (or master problem) yields a non-decreasing lower bound ($z_1^{lo} \leq z_2^{lo} \leq ... \leq z_p^{lo}$). The Outer Approximation algorithm is described as follows:

0. Initialization. Given $y^{(1)}$ and $\epsilon > 0$. Set $z^{up} = \infty$, $P = 1$, $(x^*, y^*) = \emptyset$.

1. Subproblem. Solve (sub-OA):

$$\begin{aligned} & & min\ z = f(x, y^P) \\ s.t. & & g(x, y^P) \leq 0 \\ & & x \in X \end{aligned} \qquad \text{(sub-OA)}$$

If (sub-OA) is feasible, let z^P be the optimal value of the objective function, and (x^P, y^P) its solution. Let the sets B^P and N^P be empty ($B^P = \emptyset$ and $N^P = \emptyset$). If $z^P \leq z^{up}$, set $z^{up} = z^P$, and $(x^*, y^*) = (x^P, y^P)$.

If (sub-OA) is infeasible, solve (feas-OA):

$$
\begin{aligned}
& min\ u \\
s.t. \quad & g_j(x, y^P) \leq u \quad j \in J \\
& (x, u) \in \mathbb{R}^{n+1}
\end{aligned}
\tag{feas-OA}
$$

Let (x^P, y^P) be its solution, $B^P = \{i | y_i^P = 1\}$, and $N^P = \{i | y_i^P = 0\}$.

2. Master problem. Solve (master-OA):

$$
\begin{aligned}
& min\ \alpha \\
s.t. \quad & f(x^p, y^p) + \nabla f(x^p, y^p)^T \begin{bmatrix} x - x^p \\ y - y^p \end{bmatrix} \leq \alpha && p = 1, ..., P \\
& g_j(x^p, y^p) + \nabla g_j(x^p, y^p)^T \begin{bmatrix} x - x^p \\ y - y^p \end{bmatrix} \leq 0 && j \in J^p; p = 1, ..., P \\
& \sum_{i \in B^p} y_i - \sum_{i \in N^p} y_i \leq |B^p| - 1 && B^p \cup N^p \neq \emptyset; p = 1, ..., P \\
& x \in X \\
& y \in Y
\end{aligned}
\tag{master-OA}
$$

Set $z^{lo} = \alpha$.

3. Terminate?

If $(z^{up} - z^{lo})/z^{up} \leq \epsilon$, stop with z^{up} as optimal objective value, and (x^*, y^*) optimal solution.

Otherwise, $P = P + 1$. Let y^P be the optimal the optimal value of y in (master-OA), and go back to 1.

There are two important properties of the algorithm to note:

Property 3.1 *When the set of points $p = 1, ..., P$ includes the solution of all feasible discrete variables y^p, the solution of (master-OA) and the original (MINLP) are the same.*

Proof. *The proof follows from the work by Duran and Grossmann [34].*

Property 3.1 *The Outer Approximation algorithm converges in one iteration if $f(x, y)$ and $g(x, y)$ are linear.*

Proof. *(master-OA) of an MILP is the MILP itself. Therefore, the solution of the (master-OA) of an MILP is the solution to the original MILP.*

Note that in the algorithm shown above, we solve the MILP master problem (master-OA) to optimality. This yields the best integer solution to the relaxed linear master problem. An alternative formulation for the master problem is as follows:

$$
\begin{aligned}
& min\ 0 \\
s.t.\quad & \alpha \leq z^{up}(1-\epsilon) \\
& f(x^p,y^p) + \nabla f(x^p,y^p)^T \begin{bmatrix} x-x^p \\ y-y^p \end{bmatrix} \leq \alpha \qquad p=1,...,P \\
& g_j(x^p,y^p) + \nabla g_j(x^p,y^p)^T \begin{bmatrix} x-x^p \\ y-y^p \end{bmatrix} \leq 0 \qquad j \in J^p; p=1,...,P \\
& \sum_{i\in B^p} y_i - \sum_{i\in N^p} y_i \leq |B^p| - 1 \qquad B^p \cup N^p \neq \emptyset; p=1,...,P \\
& x \in X \\
& y \in Y
\end{aligned}
\tag{R-master-OA}
$$

In (R-master-OA), the solution to the problem provides a feasible value, in which the objective (α) is smaller than the current estimate. Note that if (R-master-OA) is used instead of (master-OA), the algorithm stops when (R-master-OA) becomes infeasible instead of the optimality gap criterion ($(z^{up}-z^{lo})/z^{up} \leq \epsilon$) in (master-OA).

3.4 Generalized Benders decomposition

Generalize Bender Decomposition (GBD)[36][37] is similar to the OA method, but they differ in the linear master problem. In particular, the master problem of the GBD only considers the discrete variables $y \in Y$, and the active inequalities $J^p = \{j | g_j(x^p,y^p) = 0\}$.

Consider the constraints of the linear approximation of the convex (MINLP) at a given point (x^p,y^p), where x^p corresponds to the optimal solution of (fx-MINLP) for a given y^p:

$$
\begin{aligned}
& f(x^p,y^p) + \nabla f(x^p,y^p)^T \begin{bmatrix} x-x^p \\ y-y^p \end{bmatrix} \leq \alpha \\
& g(x^p,y^p) + \nabla g(x^p,y^p)^T \begin{bmatrix} x-x^p \\ y-y^p \end{bmatrix} \leq 0
\end{aligned}
\tag{1}
$$

Summing all of these constraints with weights μ^p, where μ^p represents the multipliers of (fx-MINLP) for the fixed y^p, we obtain (2):

$$
f(x^p,y^p) + \nabla f(x^p,y^p)^T \begin{bmatrix} x-x^p \\ y-y^p \end{bmatrix} + (\mu^p)^T \left(g(x^p,y^p) + \nabla g(x^p,y^p)^T \begin{bmatrix} x-x^p \\ y-y^p \end{bmatrix} \right) \leq \alpha \tag{2}
$$

Making use of KKT conditions, and eliminating the continuous variables x, it is possible to write the Lagrangian cut projected into the y-space as follows[38]:

$$f(x^p, y^p) + \nabla_y f(x^p, y^p)^T (y - y^p) + (\mu^p)^T \left[g(x^p, y^p) + \nabla_y g(x^p, y^p)^T (y - y^p) \right] \leq \alpha \qquad (LC^p)$$

Similarly, when (fx-MINLP) is infeasible, (feas-MINLP) is used and the following cut is obtained:

$$(\lambda^p)^T \left[g(x^p, y^p) + \nabla_y g(x^p, y^p)^T (y - y^p) \right] \leq 0 \qquad (FC^p)$$

The Generalized Benders Decomposition (GBD) algorithm follows the sames steps as the Outer Approximation. The only difference is the master MILP problem which, for the GBD, is formulated as follows:

$$\begin{aligned}
& min\ \alpha \\
s.t. \quad & f(x^p, y^p) + \nabla_y f(x^p, y^p)^T (y - y^p) + (\mu^p)^T \left[g(x^p, y^p) + \nabla_y g(x^p, y^p)^T (y - y^p) \right] \leq \alpha \quad p \in PF \\
& (\lambda^p)^T \left[g(x^p, y^p) + \nabla_y g(x^p, y^p)^T (y - y^p) \right] \leq 0 \qquad p \in PI \\
& x \in X, \alpha \in \mathbb{R}
\end{aligned}$$

(master-GBD)

where PF is the set of feasible subproblems, and PI the set of infeasible subproblems.

Note that (master-GBD) is a surrogate of (master-OA), and therefore provides a weaker linear relaxation. The advantage of GDB is that it involves only the integer variables, and it increases one cut per iteration.

3.5 Extended cutting plane

The Extended Cutting Plane (ECP) method[39] follows a similar concept as the Outer Approximation, but it avoids solving NLP subproblems. In this method, at a given solution of the master MILP (M-MIP) (x^p, y^p), all the constraints are linearized. A subset of the most violated linearized constraints is then added to the master problem. Convergence is achieved when the maximum violation lies within the specified tolerance. The algorithm provides a non-decreasing lower bound after each iteration.

The main strength of the method is that it relies solely in the solution of MILPs, for which powerful algorithms are readily available. Similarly to the OA method, it solves the problem in one iteration if $f(x, y)$ and $g(x, y)$ are linear. There are two main downsides in the algorithm. The first one is that convergence can be slow[40]. The second one is that the algorithm does not provide an upper bound (or feasible solution) until it converges.

3.6 LP/NLP based branch and bound

The LP/NLP based branch and bound method[38] is a single tree search, that can be considered as a hybrid algorithm between branch and bound, and OA. The algorithm is first initialized in a similar manner as the OA. For a given integer variable value $y^{(0)}$, problem (fx-MINLP) is solved to find a feasible solution $(x^{(0)}, y^{(0)})$. If (fx-MINLP) is infeasible, then (feas-MINLP) is solved to find the solution to the feasibility problem $(x^{(0)}, y^{(0)})$. After initializing, a linearization is performed at the initial point $(x^{(0)}, y^{(0)})$. Then the tree search starts for the master MILP, using a linear branch and bound. When a node is infeasible, it is pruned. If it is non-integer feasible, then branching is performed at that node. When a node is integer feasible, the integer variables y^p are fixed and an NLP is solved for that node (fx-MINLP). The constraints of the original MINLP are linearized at the solution of this NLP (x^p, y^p), and this linearizations are included in the master MILP. The master MILP is updated, and all the integer feasible nodes are resolved. The upper and lower bounds inferred from the tree are constantly updated, until they reach the specified tolerance. Note that, unlike traditional branch and bound, integer feasible nodes are not pruned, and they are updated every time a NLP is solved, and the new linearization included in the master MILP. Only infeasible nodes are pruned from the tree.

This method, compared to the OA, generally reduces the number of evaluated nodes, but increases the number of NLPs solved. Recent work shows that the use of modern MILP solvers and advanced MILP tools greatly improve the performance of this method[41]. The extension of this method to GBD and ECP is straightforward.

3.7 Illustrative example

The following convex MINLP is presented by Duran and Grossmann[34]:

$$
\begin{aligned}
& min\ z = 5y_1 + 8y_2 + 6y_3 + 10y_4 + 6y_5 + 7y_6 + 4y_7 + 5y_8 - 10x_3 - 15x_5 + 15x_{10} \\
& + 80x_{17} + 25x_{19} + 35x_{21} - 40x_9 + 15x_{14} - 35x_{25} + exp(x_3) + exp(x_5/1.2) \\
& - 65ln(x_{10} + x_{17} + 1) - 90ln(x_{19} + 1) - 80ln(x_{21} + 1) + 120 \\
s.t. \quad & - 1.5ln(x_{19} + 1) - ln(x_{21} + 1) - x_{14} \leq 0 \\
& - ln(x_{10} + x_{17} + 1) \leq 0 \\
& - x_3 - x_5 + x_{10} + 2x_{17} + 0.8x_{19} + 0.8x_{21} - 0.5x_9 - x_{14} - 2x_{25} \leq 0 \\
& - x_3 - x_5 + 2x_{17} + 0.8x_{19} + 0.8x_{21} - 2x_9 - x_{14} - 2x_{25} \leq 0 \\
& - 2x_{17} - 0.8x_{19} - 0.8x_{21} + 2x_9 + x_{14} + 2x_{25} \leq 0 \\
& - 0.8x_{19} - 0.8x_{21} + x_{14} \leq 0 \\
& - x_{17} + x_9 + x_{25} \\
& - 0.4x_{19} - 0.4x_{21} + 1.5x_{14} \leq 0 \\
& 0.16x_{19} + 0.16x_{21} - 1.2x_{14} \leq 0 \\
& x_{10} - 0.8x_{17} \leq 0 \\
& - x_{10} + 0.4x_{17} \leq 0 \\
& exp(x_3) - Uy_1 \leq 1 \\
& exp(x_5/1.2) - Uy_2 \leq 1 \\
& x_9 - Uy_3 \leq 0 \\
& 0.8x_{19} + 0.8x_{21} - Uy_4 \leq 0 \\
& 2x_{17} - 2x_9 - 2x_{25} - Uy_5 \leq 0 \\
& x_{19} - Uy_6 \leq 0 \\
& x_{21} - Uy_7 \leq 0 \\
& x_{10} + x_{17} + Uy_8 \leq 0 \\
& y_1 + y_2 = 1 \\
& y_4 + y_5 \leq 1 \\
& - y_4 + y_6 + y_7 = 0 \\
& y_3 - y_8 \leq 0 \\
& y \in \{0,1\}^8;\ a \leq x \leq b;\ x_3, x_5, x_9, x_{10}, x_{14}, x_{17}, x_{19}, x_{21}, x_{25} \in \mathbb{R} \\
& a^T = (0,0,0,0,0,0,0,0);\ b^T = (2,2,1,2,2,2,2,1,3);\ U = 10
\end{aligned}
\tag{3}
$$

Formulation (3) seeks to optimize the network superstructure shown in Figure 1. Table 1 shows the number of NLPs and MILPs that each of the methods requires to solve problem (3). Note that this table reflects their performance in solving example (3), and it cannot be generalized as the overall performance of

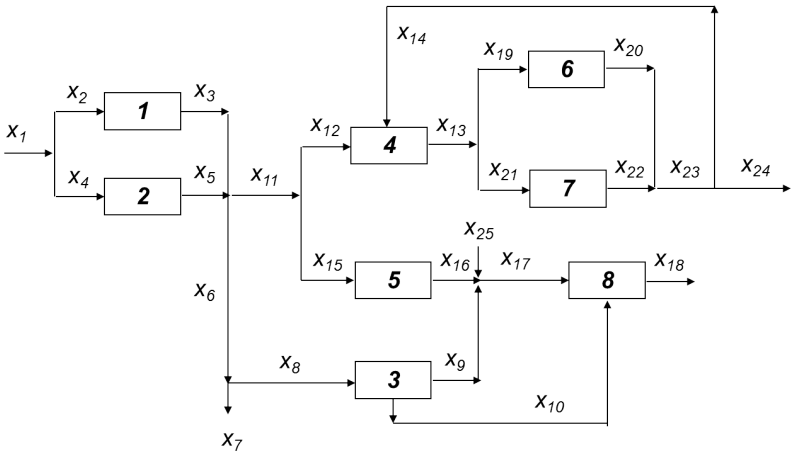

Figure 1: Network superstructure [34] for problem (3)

Table 1: Performance of convex MINLP methods for solving (3)

Method	NLP	MILP
Nonlinear branch and bound	20	-
Outer Approximation	3	3
Generalized Benders Decomposition	10	10
Extended Cutting Plane	-	15
LP/NLP branch and bound	3	13

the methods.

3.8 Convex MINLP optimization solvers

There are several convex MINLP solvers that use the different methods discussed in this section. In this section, we describe some of the main convex MINLP solvers and the methods they use. Table 2 provides a summary of this section. For a detailed review of MINLP solvers, we refer the reader to the work by Bussieck and Vigerske[42] and D'Ambrosio and Lodi[43].

α**-ECP**[44]. It is a commercial MINLP solver based on the extended cutting plane method. The solver can be applied to general MINLP problems and global optimal solutions can be ensured for pseudoconvex MINLP problems. This solver can be called through the GAMS[45] modeling system.

BONMIN (Basic Open-source Nonlinear Mixed INteger programming)[46]. It is an open source solver for solving MINLP problems. This solver includes four of the previously discussed methods: convex branch and bound(B-BB), Outer Approximation (B-OA), Extended Cutting Plane (B-Ecp), and LP-NLP branch and bound (B-QG,B-Hyb). This solver uses CBC as MILP solver (https://projects.coin-or.org/Cbc), and IPOPT[47] or Filter-SQP[48] as NLP solver. BONMIN has interfaces with AMPL, C++, GAMS and MATLAB.

Table 2: Convex MINLP solvers

Solver	MINLP solution method				
	BB	OA	GBD	ECP	LP/NLP BB
α-ECP				✓	
BONMIN	✓	✓		✓	✓
DICOPT		✓			
FilMINT					✓
MILANO	✓	✓			
MINLP-BB	✓				
MINOPT		✓	✓		
MINOTAUR	✓				
SBB	✓				

DICOPT (DIscrete and COntinuous OPTimizer)[49]. It is a commercial solver, based on the Outer Approximation algorithm. The algorithm has an interface with GAMS. It allows the selection of the MILP and NLP solvers. It contains tools that allow the handling of some nonconvex MINLP, but it does not guarantee global optimum in such cases.

FilMINT[41]. This solver is based on the LP/NLP branch and bound algorithm. The solver includes several implementation improvements, exploiting advanced tools for MILP. Some of these tools include cutting planes, preprocessing, primal heuristics, and branching and node selection strategies. Other features, particularly in the selection of a subset of linearized constraints, helps to improve the performance of the solver.

MILANO (Mixed Integer Linear And Nonlinear Optimization). This open source solver is implemented in MATLAB, and it includes a nonlinear branch and bound and an Outer Approximation algorithms. To solve the NLP, an interior point method is used, involving the warmstart scheme of Benson and Shanno[50][51].

MINLP-BB[52]. It is a commercial solver that implements nonlinear branch and bound. MINLP-BB has an interface with FORTRAN and AMPL, as well as with MATLAB through the TOMLAB optimization environment.

MINOPT[53]. It is a commercial solver that implements Generalized Bender Decomposition and Outer Approximation. MINOPT includes different LP, MILP, and NLP routines to solve MINLP.

MINOTAUR(Mixed Integer Nonlinear Optimization Toolkit: Algorithms, Underestimators and Relaxations.)(http://wiki.mcs.anl.gov/minotaur/). It is an open-source toolkit for solving convex MINLPs. It implements a nonlinear branch and bound, and has an interface with AMPL and C++.

SBB (Simple Branch and Bound). This nonlinear branch and bound is implemented in GAMS. It implements a basic branch and bound, and it allows the user to specify the NLP solver.

4 Convex GDP methods

4.1 GDP branch and bound

The idea behind the GDP branch and bound[18], which can be regarded as an extension to the disjunctive branch and bound[54], is to directly branch on the disjunctions, while considering the (r-BM) or (r-HR) of the remaining ones. The advantage of this method is that, in general, it requires the evaluation of less nodes than performing a nonlinear branch and bound over the (BM) or (HR). The number of maximum number of nodes in a traditional nonlinear branch and bound of the (BM) or (HR) is $2^{\sum\limits_{k\in K}|D_k|-1}$, while the maximum number of nodes in the GDP branch and bound is $\prod\limits_{k\in K}|D_k|$. Note that, in the extreme case in which $|D_k| = 2; \forall k \in K$ (i.e. all the disjunctions contain 2 disjunctive terms), the maximum number of nodes is the same for both branch and bound approaches ($2^{|K|}$). In cases where at least one disjunction contains more than two terms, the disjunctive branch and bound maximum number of nodes is smaller. Also, the size of the NLP (r-MINLP) decreases as the search tree progresses.

We illustrate the performance this method with the following strip packing small example[55]:

$$
\begin{aligned}
& & & min\ lt \\
& s.t. & & lt \geq x_1 + 6 \\
& & & lt \geq x_2 + 5 \\
& & & lt \geq x_3 + 4 \\
& & & \begin{bmatrix} Y_1 \\ x_1 + 6 \leq x_2 \end{bmatrix} \vee \begin{bmatrix} Y_2 \\ x_2 + 5 \leq x_1 \end{bmatrix} \vee \begin{bmatrix} Y_3 \\ y_1 - 6 \geq y_2 \end{bmatrix} \vee \begin{bmatrix} Y_4 \\ y_2 - 7 \geq y_1 \end{bmatrix} \\
& & & \begin{bmatrix} Y_5 \\ x_1 + 6 \leq x_3 \end{bmatrix} \vee \begin{bmatrix} Y_6 \\ x_3 + 4 \leq x_1 \end{bmatrix} \vee \begin{bmatrix} Y_7 \\ y_1 - 6 \geq y_3 \end{bmatrix} \vee \begin{bmatrix} Y_8 \\ y_3 - 3 \geq y_1 \end{bmatrix} \\
& & & \begin{bmatrix} Y_9 \\ x_2 + 5 \leq x_3 \end{bmatrix} \vee \begin{bmatrix} Y_{10} \\ x_3 + 4 \leq x_2 \end{bmatrix} \vee \begin{bmatrix} Y_{11} \\ y_2 - 7 \geq y_3 \end{bmatrix} \vee \begin{bmatrix} Y_{12} \\ y_3 - 3 \geq y_2 \end{bmatrix} \qquad (4) \\
& & & Y_1 \veebar Y_2 \veebar Y_3 \veebar Y_4 \\
& & & Y_5 \veebar Y_6 \veebar Y_7 \veebar Y_8 \\
& & & Y_9 \veebar Y_{10} \veebar Y_{11} \veebar Y_{12} \\
& & & 0 \leq x_1, x_2, x_3 \leq 15 \\
& & & 6 \leq y_1 \leq 10 \\
& & & 7 \leq y_1 \leq 10 \\
& & & 3 \leq y_1 \leq 10 \\
& & & Y_i \in \{True, False\};\ i = 1, ..., 12
\end{aligned}
$$

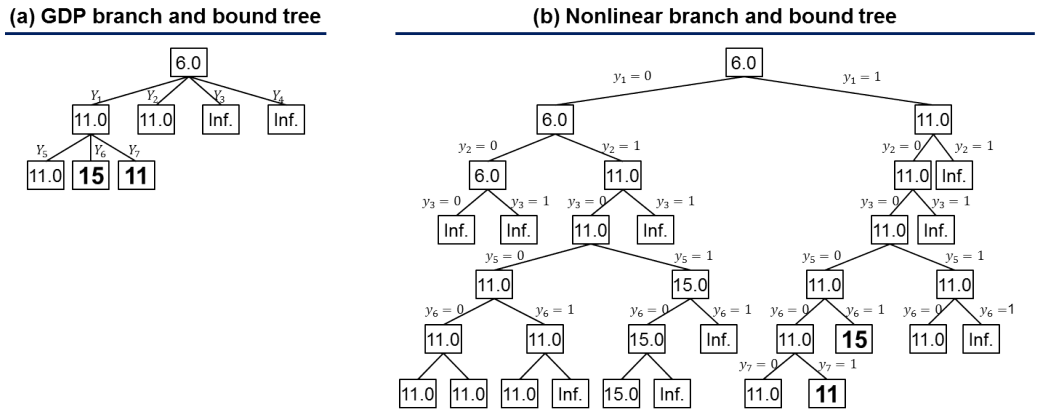

Figure 2: (a) GDP and (b) nonlinear branch and bound tree for example (4)

Note that in (4), the maximum number of nodes in the GDP branch and bound is $maxnodes = (4)(4)(4) = 64$. If the problem is reformulated using either the (BM) or (HR), and a nonlinear branch and bound is used to solve the problem, the maximum number of nodes is $maxnodes = 2^9 = 512$. Figure 2 shows the GDP branch and bound, and the convex branch and bound tree for problem (4). For both of the trees the branching selection was done in the order of the binary (and Boolean) variable. The GDP branch and bound finds the optimal solution after evaluating 8 nodes. The nonlinear branch and bound requires the evaluation of 33 nodes. It should be noted that due to the advances in MILP tools, the main MILP commercial solvers (CPLEX, GUROBI and XPRESS) are able to solve this simple problem in the root node.

4.2 Logic Based Outer Approximation

The logic based Outer Approximation[56] follows a similar idea than the Outer Approximation. It iteratively solves a master problem and a subproblem. The master linear GDP seeks to find a lower bound and a set of alternatives $Y^p_{ki} = True$. The NLP subproblem (fx-GDP) provides and upper bound. If the subproblem is infeasible, then a feasibility GDP problem is solver (feas-GDP). The solution of the subproblem (fx-GDP) or (feas-GDP) is used to perform a linearization of the constraints, which are added to the master MILP.

The initialization of the logic based Outer Approximation is different than the one of the nonlinear Outer Approximation. In particular, several NLP subproblems are solved in order to obtain a linearization of the constraints corresponding to each of the disjunctive terms $i \in D_k$ in each of the disjunctions $k \in K$. The initialization is described as follows:

1. Covering alternatives. Find a set of n valid alternatives $(KI^1, ..., KI^n)$, such that $\bigcup_{\substack{ki \in KI^p \\ p=1,...,n}} Y^p_{ki} = True$ covers $Y_{ki} = True; \forall i \in D_k; k \in K$. Selecting the smallest number of alternatives is equivalent to solving a set covering problem[56] , which is easy to solve. Set $z^{up} = \infty$, $(x^*, y^*) = \emptyset$,$p = 1$.

2. Stop initialization? If $p = n + 1$, stop initialization. Otherwise, go to step 3.

3. Initializing Subproblems. Solve (Isub-LBOA):

$$\begin{aligned} & & min\ z = f(x) & & \\ s.t. & & g(x) \leq 0 & & \\ & & r_{ki}(x) \leq 0 & \quad ki \in KI^p & \text{(Isub-LBOA)} \\ & & x^{lo} \leq x \leq x^{up} & & \\ & & x \in \mathbb{R}^n & & \end{aligned}$$

If (Isub-LBOA) is feasible, let z^p be the optimal value of the objective function, and (x^p, y^p) its solution. If $z^p \leq z^{up}$, set $z^{up} = z^p$, and $(x^*, y^*) = (x^p, y^p)$. Let the sets B^p and N^p be empty ($B^p = \emptyset$ and $N^p = \emptyset$)

If (Isub-LBOA) is infeasible, solve (Ifeas-LBOA):

$$\begin{aligned} & & min\ u & & \\ s.t. & & g_e(x) \leq u & \quad e \in E & \\ & & r_{kis}(x) \leq u & \quad kis \in KIS^p & \text{(Ifeas-LBOA)} \\ & & x^{lo} \leq x \leq x^{up} & & \\ & & (x, u) \in \mathbb{R}^{n+1} & & \end{aligned}$$

where E is the set of global constraints, and KIS^p is the set of constraints of the selected disjunctive terms KI^p. Let (x^p, y^p) be its solution. Let $B^p = \{ki | Y^p_{ki} = True\}$, and $N^p = \{ki | Y^p_{ki} = False\}$.

Set $p = p + 1$, and go to step 2.

After initializing, the logic based Outer Approximation follows the same method as the nonlinear OA:

0. Initialization. Given $(x^p, Y^p); p = 1, ..., n$, z^{up}, and (x^*, y^*) from initialization; and $\epsilon > 0$. Set $P = n$.

1. Master problem. Solve (master-LBOA) using either the (BM) or (HR) reformulation:

$$
\begin{aligned}
& min\ \alpha \\
s.t.\quad & f(x^p) + \nabla f(x^p)^T (x - x^p) \leq \alpha && p = 1, ..., P \\
& g_j(x^p) + \nabla g_j(x^p)^T (x - x^p) \leq 0 && j \in J^p; p = 1, ..., P \\
& \underset{i \in D_k}{\vee} \begin{bmatrix} Y_{ki} \\ r_{ki}(x^p) + \nabla r_{ki}(x^p)^T (x - x^p) \leq 0 \end{bmatrix} && k \in K \\
& \underset{ki \in N^p}{\vee} Y^p_{ki} \underset{ki \in B^p}{\vee} \neg Y^p_{ki} && B^p \cup N^p \neq \emptyset; p = 1, ..., P \quad \text{(master-LBOA)} \\
& \underset{i \in D_k}{\veebar} Y_{ki} && k \in K \\
& \Omega(Y) = True \\
& x^{lo} \leq x \leq x^{up} \\
& x \in \mathbb{R}^n \\
& Y_{ki} \in \{True, False\} && k \in K, i \in D_k
\end{aligned}
$$

Set $z^{lo} = \alpha$.

2. Terminate?

If $(z^{up} - z^{lo})/z^{up} \leq \epsilon$, stop with z^{up} as optimal objective value, and (x^*, y^*) optimal solution.

Otherwise, $P = P + 1$, Let y^P_{ki} be the optimal the optimal value of y_{ki} in the MINLP reformulation of (master-OA). Set $Y^P_{ki} = True$ if $y^P_{ki} = 1$, $Y^P_{ki} = False$ otherwise. $KI^P = \{ki | Y^P_{ki} = True\}$. Go to step 3.

3. Subproblem. Solve (sub-LBOA):

$$
\begin{aligned}
& min\ z = f(x) \\
s.t.\quad & g(x) \leq 0 \\
& r_{ki}(x) \leq 0 && ki \in KI^P \quad \text{(sub-LBOA)} \\
& x^{lo} \leq x \leq x^{up} \\
& x \in \mathbb{R}^n
\end{aligned}
$$

If (sub-LBOA) is feasible, let z^P be the optimal value of the objective function, and (x^P, y^P) its solution. If $z^P \leq z^{up}$, set $z^{up} = z^P$, and $(x^*, y^*) = (x^P, y^P)$. Let the sets B^P and N^P be empty ($B^P = \emptyset$ and $N^P = \emptyset$)

If (sub-LBOA) is infeasible, solve (feas-LBOA):

$$
\begin{aligned}
& & & min\ u & & \\
& s.t. & & g_e(x) \leq u & & e \in E \\
& & & r_{kis}(x) \leq u & & kis \in KIS^P \\
& & & x^{lo} \leq x \leq x^{up} & & \\
& & & (x, u) \in \mathbb{R}^{n+1} & &
\end{aligned}
\qquad \text{(feas-LBOA)}
$$

where E is the set of global constraints, and KIS^P is the set of constraints of the selected disjunctive terms KI^P. Let (x^P, y^P) be its solution. Let $B^P = \{ki|Y^P_{ki} = True\}$, and $N^P = \{ki|Y^P_{ki} = False\}$. Go back to step 1.

It is important to note that the logic based Outer Approximation requires the solution of more NLP problems to initialize. However, the subproblems in the initialization (Isub-LBOA), (Ifeas-LBOA), and in the main algorithm (sub-LBOA), (feas-LBOA) are smaller NLP problems than the ones in the nolninear OA, since they only contain a subset of the constants. Also, the logic based Outer Approximation sometimes requires fewer iterations of the master problem to converge[21].

5 Nonconvex MINLP

Noncomvex MINLP is one of the most complex and active fields in optimization. The accurate modelling of many real problems, particularly in chemical engineering, require the use of nonconvex constraints[16, 25, 57, 58]. In this section we present the most common deterministic method to solve nonconvex MINLP, the spatial branch and bound. In order to explain the algorithm, we first present two main concepts that are used in most applications: Relaxations of factorable formulations [16, 59, 60], and bounds tightening[61–64]. For a more comprehensive description of deterministic and heuristic methods, we refer the reader to the books[16, 57, 65] or review papers on global optimization[14, 66, 67]

5.1 Relaxations of factorable formulations

In many MINLP problems, the objective function and the constraints can be reformulated by taking recursive sums and products of univariate functions. These problems are called factorable formulations [16, 68].

It is possible to reformulate (via symbolic manipulation) a factorable MINLP into a standard form. The purpose of the standard form is to isolate the nonlinear terms in order to address them more easily. It is important to note that there are many alternative standard forms, depending on the algorithm and application. Isolating purely univariate functions is the extreme case of standard form. We illustrate alternative standard forms with the following constraint:

$$(x_1^3 + x_1^2 + x_1x_2)^{1/3} \leq 0 \tag{Constr}$$

The following three reformulations are alternative standard forms of (Constr).

$$\begin{aligned} (z_1)^{1/3} &\leq 0 \\ z_1 &= z_2 + z_3 \\ z_2 &= x_1^3 + x_1^2 \\ z_3 &= x_1x_2 \end{aligned} \tag{Constr.A}$$

An alternative standard form could be:

$$\begin{aligned} (z_1)^{1/3} &\leq 0 \\ z_1 &= z_2 + z_3 + z_4 \\ z_2 &= x_1^3 \\ z_3 &= x_1^2 \\ z_4 &= x_1x_2 \end{aligned} \tag{Constr.B}$$

Finally, assuming $x_1, x_2 > 0$, it is possible to reformulate (Constr) with purely univariate functions as follows:

$$\begin{aligned} (z_1)^{1/3} &\leq 0 \\ z_1 &= z_2 + z_3 + z_4 \\ z_2 &= x_1^3 \\ z_3 &= x_1^2 \\ z_5 &= z_6 + z_7 \\ z_5 &= log(z_4) \\ z_6 &= log(x_1) \\ z_7 &= log(x_2) \end{aligned} \tag{Constr.C}$$

(Constr.C) is the extreme case in which each nonlinear constraint contains an univariate function.

In most application and solvers, the standard form is a combination of mainly univariate, bivariate functions, and some particular multivariate functions with specific form[16, 59, 60].

Once the problem is in standard form, it is possible to develop a convex relaxation of the nonconvex constraints. This relaxation is achieved by using convex constraints that over estimate the feasible region of the nonconvex constraints. In many algorithms these approximations tend to be polyhedral[69]. Note that the standard form of a particular algorithm will depend on the library of multivariate functions for which over and under estimators have been developed. We illustrate this relaxation with the well-known McCormick

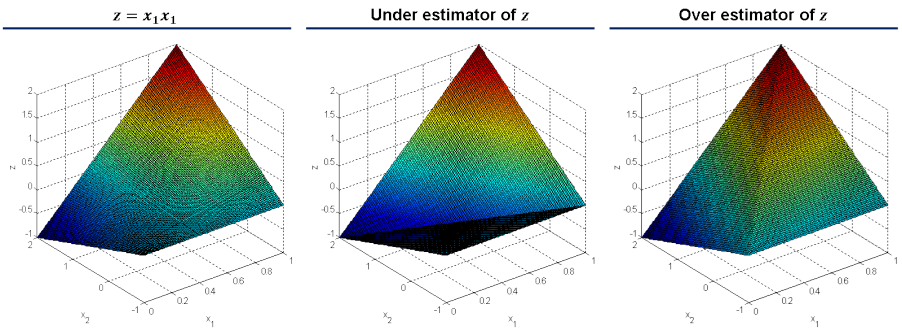

Figure 3: McCormick over and under estimators for bilinear term

over and under estimators of a bilinear term[59]. The convex (linear) envelopes of $z = x_1 x_2$ are as follows:

$$\begin{aligned} z &\geq x_1^{lo} x_2 + x_2^{lo} x_1 - x_1^{lo} x_2^{lo} \\ z &\geq x_1^{up} x_2 + x_2^{up} x_1 - x_1^{up} x_2^{up} \\ z &\leq x_1^{lo} x_2 + x_2^{up} x_1 - x_1^{lo} x_2^{up} \\ z &\leq x_1^{up} x_2 + x_2^{lo} x_1 - x_1^{up} x_2^{lo} \end{aligned} \tag{5}$$

(5) is a convex (polyhedral) relaxation of the bilinear term $z = x_1 x_2$. Figure 3 illustrates the McCormick under and over estimators of a bilinear term. It is easy to see that the feasible region that lies between the over and under estimators of z overestimates the actual feasible region of z. A high-level description for different multivariate relaxations can be found in the review paper by Floudas[67].

From (5), it is clear that the convex relaxation is strongly dependant on the bounds of the variables. This is not only the case of bilinear terms. In general, the tighter the bounds of the variables are, the better the convex approximation will be. Therefore, to improve the convex relaxation, domain reduction (bound tightening) techniques are usually applied.

5.2 Bound Tightening

As described in the previous section, tightening the bound of the variables helps to improve the approximation of the convex relaxation. Note that it is possible to solve the following MINLP to improve the lower bound of x_i variable:

$$
\begin{aligned}
& & & min\ x_i \\
& s.t. & & f(x,y) \leq z^{up} \\
& & & f(x,y) \geq z^{lo} \\
& & & g(x,y) \leq 0 \\
& & & x \in X \\
& & & y \in Y
\end{aligned}
\tag{6}
$$

where z^{up} and z^{lo} are upper and lower bounds of the objective function (if no bounds are known, consider $z^{lo} = -\infty$ and $z^{up} = \infty$).

Solving (6) for every variable yields better lower bounds. Solving (6) as a maximization problem yields better upper bounds for the variables. However, (6) is computationally very expensive, and it is impractical to solve several MINLPs, with a similar complexity to the original MINLP, to improve the bounds of the variables. In most applications alternative techniques are used[70]. The two most common type of methods for domain reduction are Feasibility-Based Bound Tightening and Optimality-Based Bound Tightening.

5.2.1 Feasibility-Based Bound Tightening

This type of methods evaluates each constraint with the current variable bounds, and tries to improve bounds by maintaining feasibility in the constraint. A recent method by Belotti[71] uses pairs of constraints instead of individual constraints to infer bounds. Different techniques have been developed to infer bounds on MILP problems[62, 63], and on MINLP problems[64, 72]. To illustrate this method consider the constraint $x_3 = x_1 x_2$, where the variable bounds are $x^{lo} = (2, 1, 0)^T$ and $x^{up} = (4, 4, 4)^T$. In order to find a better lower bounds for x_3, it is possible to use the bounds of the other variables, so $x_3 \geq x_1^{lo} x_2^{lo}$ or $x_3 \geq (2)(1)$. It is clear from this equation that a better lower bound for x_3 is $x_3^{lo} = 2$. Also, an improved upper bound for x_2 can be calculated using $x_3^{up} \leq x_1^{lo} x_2$ or $4 \leq (2)x_2$. It can be implied then that $x_2^{up} = 2$. After this iteration the method obtains the improved bounds $x^{lo} = (2, 1, 2)^T$ and $x^{up} = (4, 2, 4)^T$. It is important to note that the sequence in which we select the variables will impact the values of the improved bounds. Also, this method can be applied repetitively to iteratively improve bounds. This method is computationally cheap, but the convergence might be slow.

5.2.2 Optimality-Based Bound Tightening

This method follows the same idea described at the beginning of this section, but instead of minimizing (maximizing) a variable over the original feasible region of the MINLP, it is minimized (maximized) over the feasible region of the convex continuous relaxation. Consider the following two constraints $x_3 = x_1 x_2$, and

$x_1^2 = x_2$, with bounds $x^{lo} = (1, 1, 2)^T$, $x^{up} = (4, 4, 4)^T$. The formulation to minimize the bound of x_1 over the convex NLP relaxation of the constraints is as follows:

$$\begin{aligned} & min\ x_1 \\ s.t. \quad & x_3 \geq x_1 + x_2 - 1 \\ & x_3 \geq 4x_2 + 4x_1 - 16 \\ & x_3 \leq x_2 + 4x_1 - 4 \\ & x_3 \leq 4x_2 + x_1 - 4 \\ & x_1^2 \leq x_2 \\ & x_1^2 \geq -4 + 5x_1 \\ & 1 \leq x_1 \leq 4 \\ & 1 \leq x_2 \leq 4 \\ & 2 \leq x_3 \leq 4 \end{aligned} \tag{7}$$

The solution of (7) provides $x_1^{lo} = 1.11$. By maximizing x_1 in (7), the bound $x_1^{up} = 1.79$ is found. The same problem, minimizing and maximizing x_2 with the new bounds for x_1 yields $x_2^{lo} = 1.35$ and $x_2^{up} = 3.5$.

This method also depends on the sequence in which the variables are selected, and it can be applied iteratively. Note that Feasibility-Based Bound Tightening is much faster, but it evaluates one, and sometimes two constraints at a time. Optimality-Based Bound Tightening captures the tradeoff between the convex relaxation of all of the constraints.

5.3 Spatial branch and bound

The spatial branch and bound algorithm is the most common deterministic method for solving nonconvex MINLP problems. The method requires a convex relaxation of the MINLP (for example, described at the beginning of Section 5). To describe this method we will consider a nonconvex NLP. The integrallity can be tackled by branching in the integral variable that obtained a fractional value, in a similar manner as the branch and bound method described in Section 3.2. The spatial branch and bound first solves the convex relaxation of the NLP, which provides a lower bound. Afterwards, it seeks to solve the NLP with a local solver. If the NLP is feasible, it provides an upper bound. If it is infeasible, then the upper bound is considered ∞. If the gap between upper and lower bound is within tolerance, the algorithm stops with an optimal solution. If not, it branches on the domain of a variable, yielding two new NLPs. Each NLP is solved to find upper and lower bound as described earlier. The lowest upper bound in the new regions provides an upper bound for the problem. This method is repeated until the gap between the upper and lower bound lies within the tolerance. If the lower bound of a region is larger than the upper bound of the problem, the region is then fathomed.

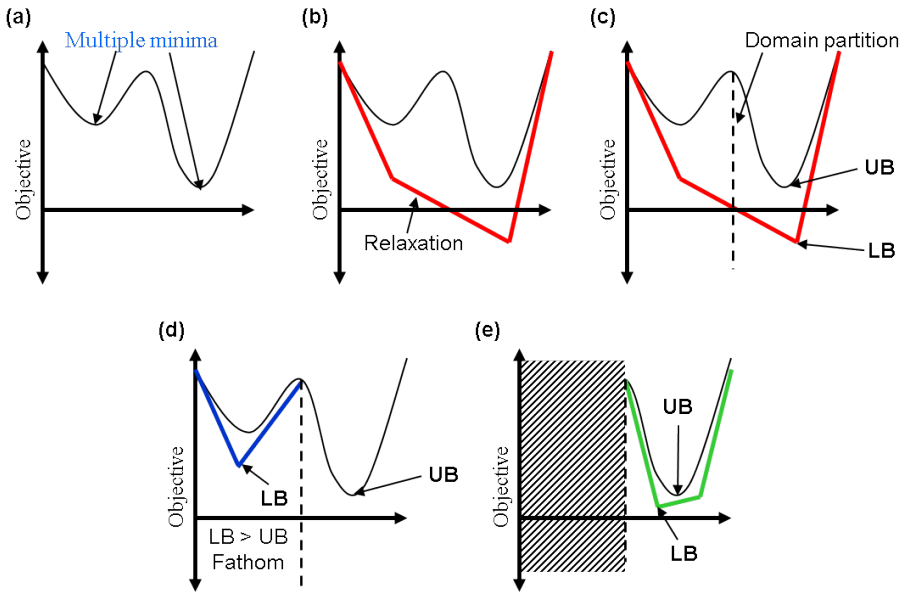

Figure 4: Spatial branch and bound for a nonconvex NLP. (a) Nonconvex function with multiple minima. (b) Convex (polyhedral) relaxation of the objective function. (c) Upper and lower bound in the total region, and partition of domain into two subregions. (d) Lower bound of the subregion is larger than known upper bound, so the subregion is fathomed. (e) Gap between upper bound and lower bound of the second subregion is within tolerance, optimal solution is found.

Figure 4 illustrate the spatial branch and bound in a nonconvex function. Figure 4.a shows an objective function with two local minima. Figure 4.b shows the convex (polyhedral) relaxation of the function. Figure 4.c illustrates the upper bound that is found by solving a local NLP, and the lower bound that is found by solving the relaxed convex NLP. It also shows the partition of the domain into two subregions. Figure 4.d shows that the relaxation of the left subregion provides a lower bound that is larger than the best known upper bound. Therefore, the subregion can be fathomed. Figure 4.e shows that gap between the lower bound of the subregion and the upper bound of the problem lies within the tolerance, so the algorith stops with an optimal solution.

5.4 MINLP global solvers

In this section we describe some of the deterministic global solvers. For a detailed review of MINLP solvers, we refer the reader to the work by Bussieck and Vigerske[42] and D'Ambrosio and Lodi[43].

α**BB** (α-based branch and bound)[73]. For the relaxation, this solver replaces nonconvex terms of special structure with customized convex lower bounding functions. It replaces nonconvex terms with generic structures using a quadratic under estimator (the α parameter method) to generate the convex under estimators.

ANTIGONE (Algorithms for coNTinuous / Integer Global Optimization of Nonlinear Equations)[74]. This global solver was preceded by APOGEE. In the relaxation of the bilinear terms, this algorithm uses

piecewise-linear relaxation, which involves mixed integer linear constraints rather than continuous convex constraints. ANTIGONE has an interface with GAMS and AIMMS.

BARON (Branch And Reduce Optimization Navigator)[75]. This commercial solver uses both linear and nonlinear convex relaxations for nonconvex terms. The solver also includes tools for detecting convexity and various domain reduction techniques. BARON has an interface with GAMS and AIMMS.

COUENNE (Convex Over and Under ENvelopes for Nonlinear Estimation)[76]. This solver is open source. It implements a similar method to BARON, using some advanced tools for domain reduction. COUENNE has an interface with GAMS and AIMMS.

LindoGlobal[77]. Similar method to BARON and COUENNE. It has interface with GAMS.

SCIP (Solving Constraint Integer Programs)[78]. It is a non-commercial solver that follows a similar approach to BARON, COUENNE and Lindo Global. It contains advanced tools for the solution of MILP, including constraint programming techniques. SCIP has interfaces with AMPL, GAMS, MATLAB, OSiL and ZIMPL.

6 Solution strategies for nonconvex GDP

In this section we first present a technique to improve the Optimality-Based Bound Tightening technique. We then present an example which is solved heuristically using a logic based Outer Approximation. This example shows considerable reduction in solution times when using a GDP method.

6.1 Improving bound tightening through basic steps

Ruiz and Grossmann[79, 80] propose improving the Optimality-Based Bound Tightening technique by strengthening the continuous relaxation of the convex approximation. This is achieved by using a logic operation called basic step[81], which is the intersecting between two disjunctions. For a more comprehensive review of basic steps in GDP, and their equivalence to basic steps in disjunctive programming, we refer the readers to the work by Sawaya and Grossmann[55] and by Ruiz and Grossmann [82].

Figure 5 illustrates tightness of relaxation of the (HR) reformulation of a simple GDP, before and after intersecting the disjunctions. The illustration shows a feasible region described by two disjunctions with two disjunctive terms $(([A_1] \vee [A_2]) \wedge ([B_1] \vee [B_2]))$. Figure 5.a shows the continuous relaxation of the (HR) reformulation. Figure 5.b shows that, after a basic step, the two disjunctions are intersected to form a new single disjunction $([A_1] \wedge [B_1]) \vee ([A_2] \wedge [B_2])$. It is easy to see from the figure that the basic step improves the tightness of the relaxation.

Basic steps improve the tightness of the continuous relaxation of the (HR) reformulation, but increase the problem size. There has been work suggesting which basic steps to apply[55, 81, 82], and to algorithmically

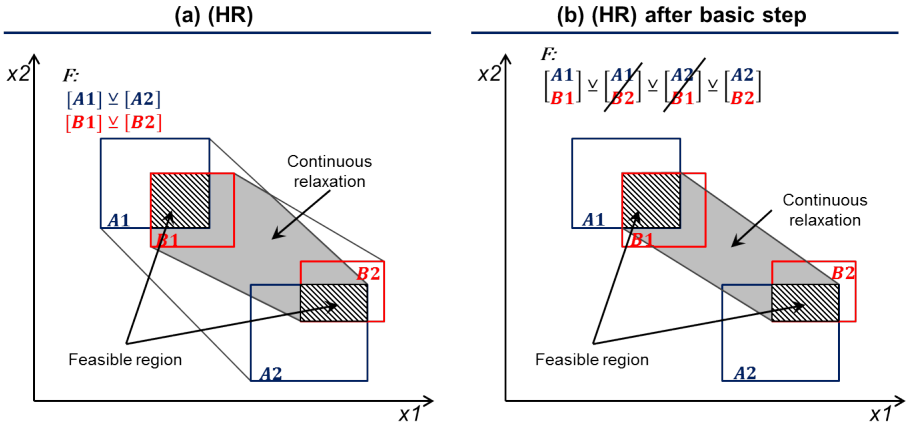

Figure 5: Illustration of (HR) (a) before, and (b) after the application of a basic step

Table 3: Bound tightening by using basic steps in GDP models

Instance	Bounding(%) using (HR)	Bounding(%) using (HR) after basic steps
Instance 1[79]	48	100
Instance 2[79]	48	100
Instance 3[79]	48	100
Instance 4[79]	48	100
Example 1[79]	35	38
Example 2[79]	33	33
Example 3[79]	85	99
Example 4[79]	8	16
Example 5[79]	1	1
Example 6[79]	98	99
Procnet 1[80]	51.3	67.0
Procnet 2[80]	40.5	47.2
RXN 1[80]	51.0	66.0
RXN 2[80]	51.0	66.0
HEN 1[80]	13.8	35.0
HEN 2[80]	7.5	97

improve formulations while limiting the problem size growth [83].

By tightening the continuous relaxation of the convex approximation of the original nonconvex GDP, the bounds obtained through the Optimality-Based Bound Tightening technique are improved. Consider $Bounding(\%) = 100(1 - \sum_i (x_i^{up^*} - x_i^{lo^*})/(|I|(x_i^{up} - x_i^{lo})))$, where $i \in I$ are the variables, x_i^{up} and x_i^{lo} are the original variable bounds and $x_i^{up^*}$ and $x_i^{lo^*}$ are the variable bounds after bound tightening. Ruiz and Grossmann[79, 80] show the improvement in bound tightening of using basic steps in several instances, as summarized in Table 3. It is easy to see from Table 3 that the bound tightening technique can be greatly improved in some instances by using basic steps.

Table 4: Performance of MINLP and GDP heuristic methods

	Optimal solution ($/year)		Solution time (s)	
Instance	OA	LBOA	OA	LBOA
Example 1: n-pentane/n-hexane/n-heptane	61,280	51,520	323	14
Example 2: butane/toluene/o-xylene	79,962	80,720	648	211
Example 3: methanol/ethanol/water	116,320	117,600	501	400

6.2 Improving Outer Approximation solution times using GDP

Barttfeld el at[84] solved different instances to find the optimal design of a multicomponent distillation column, using equilibrium equations. The MINLP and GDP formulations they presented were solved using a heuristic method to find the minimum design cost. For the MINLP there was a presolving stage, and then it was solved using Outer Approximation. The GDP representation was solved using a preprocessing stage, and then a logic based Outer Approximation.

Table 4 summarizes the performance of both heuristic methods. The logic based Outer Approximation provides a better solution in the first instance, but slightly worst in instances 2 and 3. It is important to note that the solution times of the logic based OA are smaller than the ones of the Outer Approximation. The main reason for the improvement in the solution time is that the subproblems in the logic based OA are smaller NLPs, since they only consider the constraints selected in the master problem as described earlier in Section 3. Not many applications of GDP methods for nonconvex problems can be found in the literature, but this work illustrates some of their potential.

7 MINLP and GDP applications in PSE

In this section we will present some of the applications of MINLP in PSE. It is important to consider that one of the main concepts in optimization in PSE is the idea of superstructure. A superstructure contains most (or all) of the alternatives a system can have. Superstructures are defined by the modeller, and much of the research in PSE includes the definition of alternative superstructures. The mathematical optimization approach will seek to find the optimal of the proposed alternatives. For a more comprehensive description on superstructures, we refer the reader to the work by Grossmann et al.[1].

7.1 Process synthesis

In this section, we present a summary some of the main applications of MINLP and GDP in process synthesis. Grossmann et al.[1] provide a more comprehensive review on the development of optimization models and methods in process synthesis.

Process Flowsheet Synthesis. MINLP has been widely used for process flowsheet synthesis. The problem seeks to obtain the optimal configuration of a process in a given superstructure[85–87]. The superstructure contains alternative units, interconnections, and process properties. The selection of units and their interconnections are modelled using binary variables, while the properties of the process (flow, concentration, etc.) are modelled with continuous variables. Alternative superstructure representations of processes have also been proposed[88–91]. Formulation (3), presented in Section 3.7, is a simple example of a process flowsheet superstructure and then MINLP which represents it.

Process flowsheet synthesis is one of the areas where GDP has been most successful. Raman and Grossmann[92] proposed a GDP model for process flowsheet synthesis. Disjunctive programming techniques for the optimization of process systems with discontinuous investment costs-multiple size regions were presented by Turkay and Grossmann[93]. Turkay and Grossmann[94] also presented logic-based MINLP algorithms for the optimal synthesis of process networks. Later, Yeomans and Grossmann[95] formulated the two fundamental superstructures for process systems (state task network and state equipment network) with GDP. GDP methods have shown to improve solution times for linear, convex and nonconvex process flowsheet synthesis problems[55, 80, 82, 95, 96].

The accurate modelling of a process flowsheet, including the detailed formulations of each unit, normally yields a large scale nonconvex MINLP. There are two general frameworks that are normally used to tackle these problems: decomposition techniques[97–100], and surrogate models[101–104]. There are many processes in which flowsheet optimization has been applied over the last 30 years. The most recent work has focused mainly on bioenergy systems and biorefineries[105–114], polygeneration systems[115–117], and gas-to-liquid complexes[118].

Reactor Network. This problem seeks to optimize the configuration of a reactor system for a given feed and a given set of reactions. Two main mathematical programming approaches exist for this problem: superstructure optimization and targeting. A review of both methods is provided by Hildebrandt and Biegler[119].

Superstructure optimization methods require the modeller to postulate a superstructure, which represents possible configurations for the reactor network. This structure is formulated as an MINLP and solved with optimization Tools. Achenie and Biegler[120] postulated a superstructure NLP model. The continuous model allows them to select network structure, reactor type, and the amount of heat addition. This model used a dispersion coefficient to determine the reactor type. Later, they postulated an alternative NLP which used the recycle ratio as the determinant of reactor type[121]. Kokossis and Floudas[122–124] present a superstructure of CSTRs and PFRs. In the MINLP, the PFRs are modelled as a series of CSTRs, which eliminates the use of differential equations in the model. Smith and Pantelides[90] presented a reaction and separation network with detailed unit operation models. Esposito and Floudas[125] make use of the global optimization tools to solve a nonconvex MINLP in which the PFRs of the superstructure are modelled using differential algebraic

equations.

The targeting method seeks to identify the maximum possible performance. A reactor network that meets this criteria is then determined. Horn[126] introduced the concept of "attainable region". Attainable region is the convex hull of concentrations for a given feed and reaction scheme. The geometrical concepts that allow the derivation of the attainable region, and further extensions to higher dimensions where developed mainly by Glasser, Hildebrandt, and Feinberg[127–130]. There are two main downsides of the "attainable region" technique. The first one is that, since it is a graphic method, it can handle at most three dimensions. The second one is that every time that the conditions for the reactor problem change, the region must be recalculated. To solve this issues, and improve the performance of the method, Biegler et al.[131–135] developed hybrid methods that combines "attainable region" with optimization techniques.

Single Distillation columns. The simplest distillation design problem is to select the feed tray in a distillation column with a given number of trays. The superstructure of this problem was postulated by Sargent and Gaminibandara[136] for ideal mixtures, and extended later for azeotropic cases[137]. Viswanathan and Grossmann[138] presented the superstructure of a distillation column in order to optimize not only the feed tray, but also the number of trays in the column. The model also allowed the possibility of multiple feeds. This model was later incorporated into more comprehensive superstructures which incorporated thermodynamic components[90, 139], and thermally coupled distillation columns and dividing wall columns[140].

One of the main issues of the tray by tray MINLP model of a column is the large number of nonconvex constraints in the model. Yeomans and Grossmann[141] proposed a GDP model for the tray by tray representation of the column, as well as for the sequence superstructure. They proposed a logic based Outer Approximation approach, which considerably reduces the number of nonconvex constraints in each iteration. Based on the tray by tray GDP model, Barttfeld et al.[84] presented a computational comparison between the MINLP and GDP models using a heuristic algorithm, showing the advantages in solution times of the GDP model.

Distillation sequences. The separation problem was originally defined more than 40 years ago by Rudd and Watson[142]. For the MINLP and GDP applications, Yeomans and Grosmann[95] characterized two major types of superstructure representation: State-Task-Network (STN) and State Equipment Network (SEN). They described both representations using GDP models.

For sharp split columns, the superstructure can be modelled as an MINLP. The GDP formulation for the STN model is straight forward[143]. In the case where each column is assigned to each of the split separation tasks, the model reduces to the MILP superstructure proposed by Andrecovich and Westerberg[144]. The SEN structure is not as straight forward, but it is numerically more robust[141, 143]. Novak et al[145] used this representation before being formalized by Yeomans and Grosmann[95].

In order to develop more efficient distillation sequences, heat integration between different separation

tasks can be considered. Paules and Floudas[146, 147] proposed and MINLP model for heat integrated distillation sequences, and Raman and Grossmann[92] a disjunctive representation.

Thermally coupled systems seek to reduce the inherited inefficiencies due to the irreversibility during the mixing of streams at the feed, top, and bottom of the column. The design and control of these columns are complicated, but they were made possible by the design concept of "thermodynamically equivalent configuration"[148–151], and the improvement in control strategies[152–156]. For zeotropic mixtures, Agrawal[157] characterized a subset of the possible configurations named basic configurations. This concept helps to reduce the search space of feasible configurations, since non-basic configurations normally have higher overall costs[158–160]. Algorithms for finding the basic configurations[103], and mathematical representations of these configurations have been proposed[161, 162]. In terms of GDP, logic rules that include all the basic column configurations have been developed by Caballero and Grossmann[158, 163, 164].

Heat exchange networks. A comprehensive heat exchange network (HEN) review with annotated bibliography is provided by Furman and Sahinidis[165]. A more recent review on the developments of HEN is provided by Morar and Agachi[166]. Following the classification proposed by Furman and Sahinidis[165], the work in HENs is divided in sequential or simultaneous synthesis methods.

The sequential synthesis method decomposes the design of HEN typically in three simpler sequential problems, but it does not guarantee the global optimum of the HEN. The first problem seeks to minimize the utility costs. The second one seeks to minimize the number of heat exchange units, while satisfying the minimum utility cost previously found. The last problem minimizes the network cost subject to the minimum number of units found. The first problem (minimizing utility usage/cost) can be an LP[167–169], an MILP or an MINLP[170–172]. The second problem is typically an MILP[167, 168, 173, 174]. The most common model for the third problem that minimizes the network cost is an NLP[175].

In simultaneous synthesis the HEN is optimized without decomposing the problem. One of the first MINLP models was presented by Yuan et al.[176], describing a superstructure of the network. This MINLP, however, does not allow the splitting or mixing of streams. The work by Floudas and Ciric[177–180] presents an MINLP model that optimizes all of the costs of a HEN without the need of decomposition. Yee and Grossmann[181, 182] presented an MINLP formulation of a multi-stage superstructure that allows any pair of hot and cold streams to exchange heat in every stage. This model was extended to include flexibility[183–185], detailed exchanger design models[186, 187], and isothermal streams involving phase change[188].

The simultaneous optimization of process flowsheet and heat integration was addressed by Duran and Grossmann[189]. Particular applications of heat integration with optimization of distillation columns and distillation sequences where addressed earlier in this section. Grossmann et al.[190] presented a GDP model for simultaneous flowsheet optimization and heat integration. Recently, Navarro-Amoros et al.[191] extended this GDP model for heat integration with variable temperatures.

Utility systems. Petroulas and Reklaitis[192] proposed a mathematical optimization model for the design of utility systems, based on an LP and dynamic programming approach. An MILP model was formulated by Papoulias and Grossmann[193], which assumed linear capital costs with fixed charge and linear energy balance. Bruno et al.[194] proposed a refined version of this model by including nonlinear functions, presenting the first MINLP model for the design of utility systems. This MINLP introduced nonlinearities to represent the cost of equipment and the plant performance in terms of enthalpies, entropies and efficiencies. The model considers steam properties at specific pressures, so it cannot simultaneously optimize the operating conditions of the steam levels. Savola et al.[195] presented a modified MINLP model that allows the simultaneous optimization of pressure levels, by using correlations that depend on both pressure and temperature.

Water Network. A very comprehensive review of water network design methods with literature annotations is provided by Jezowski[196]. The optimization based methods for water network design are based on superstructures, which encompasses most (or all) feasible solutions for the network. There are four types of water networks normally modelled. The network of water-using processes is called water-using network (WUN). The second type is the wastewater treatment network (WWTN). These two networks are subsystems of the third type of network: the total water network (TWN)[197, 198]. Finally, the fourth type is the complete water network (CWN), which is a TWN with pretreatment units[199]. The models in water networks typically seek to optimize the configuration of the network and units operating conditions, satisfying the water quantity and quality requirements in the WUN, with water discharges within the environmental restrictions.

The superstructure based optimization of the WUN is relatively simple, due to the fixed number of water using units. However, the use of alternative water treatment units in WWTN, TWN, and CWN makes the problem more complicated. In particular, the mixing of streams yields bilinear terms (which are nonconvex), and the operating constraints in each of the units may be highly nonconvex. In addition the selection of units and interconnections are represented by binary variable. The combination of these constraints and variables result in a nonconvex MINLP. The WWTN alone has been shown to be a very difficult problem, equivalent to the generalized pooling problem[200].

The first MINLP model for the simultaneous optimization of water networks was presented by Papalexandri et al.[201]. An alternative representation was proposed by Szitkai, et al.[202], presenting a water network modified version of the HEN superstructure of Yee and Grossmann[182]. Since the MINLP that arises in water networks is difficult to solve, five strategies are normally used for solving the problem. The first one is the linearization of constraints, typically used for the WUN. The first linearization was developed for a single-contaminant problem[203], and its optimality constraints by Savelski and Bagajewicz[204]. The model was later extended for multiple contaminants[205], thought the conditions are different[206], and it requires a specialized branching approach. Other linearization techniques for WUN with regeneration processes[207], near-optimal single stage method for WUN with multi-contaminant[208], and WWUN[209] have been pre-

sented. The second approach is to find good solutions (locally optimal) through good initial points. A common practice is to fix the outlet concentrations at the maximum values[210, 211]. The third technique is sequential optimization, originally presented by Takama[212]. Some of the sequential optimization approach methods include: a relaxed NLP model[213], dividing and sequentially reducing concentration intervals for WUN[214], an MILP-LP technique for TWN[215], and an LP-NLP approach for the WWTN[216]. The fourth strategy is to solve the MINLP problem by using meta-heuristic algorithms[217–219]. The last approach is to solve the MINLP problem using global optimization techniques[220–222].

Dong et al.[223–227] have recently recent proposed MINLP models that incorporate the design of heat exchange networks with water networks, based on the state-space superstructure proposed by Bagajewicz et al.[228, 229]. Ruiz and Grossmann[79] present a simplified GDP formulation of the WWTN, as an illustration of the bound tightening technique.

7.2 Planning and Scheduling

Process planning and scheduling has been an area of much interest for industry and research over the past decades[2, 230–232]. Although the boundary between planning and scheduling is sometimes nor clear, time scale provides a distinction between the two. Planning normally involves medium-term (e.g. year, quarters, months or weeks) decisions such as assignment of production tasks to facilities and transportation logistics. Scheduling defines short-term decisions such as assignment of tasks to units, sequencing of tasks in each unit, and determining starting and ending times for the execution of tasks. A review of planning and scheduling in the process industry is provided by Kallrath[231]. Models and techniques to integrate planning and scheduling have also receive attention in recent years[233]. In this section we present some of the main MINLP applications for planning and for scheduling.

Planning. Planing models are usually not as detailed as scheduling models, so most planing models are represented using MILP formulations[230]. However, refinery planning is one of the few areas where nonlinear models have been used. Moro et al.[234] presented an NLP planning models for refinery diesel production, extending it later to petroleum refineries[235]. Zhang at al.[236] presented one of the first MINLP models for refinery planing, which simultaneously optimizes liquid, hydrogen, and steam and power flows. They deal with the nonlinear terms by using picewise linear approximations. Elkamel et al.[237] presented an MINLP multiobjective model for the refinery planning, seeking to maximize profit while reducing CO2 emissions. Neiro and Pinto[238] presented an MINLP implementation for multiperiod refinery planning models using empirical relations. Alattas et al.[239] presented a multiperiod MINLP model that utilizes a more precise representation of the crude distillation unit, using the model introduced by Geddes[240]. Shah et al.[162] provide a literature review on existing approaches that address the problem of enterprise-wide

optimization in the petroleum industry.

Scheduling. Many scheduling models assume process simplifications, and can be modelled as MILP[241–243]. Castro and Grossmann[244] presented the traditional MILP formulations as a derivation of GDP models. Many applications, however, require the use on nonlinear constraints to better represent the process.

There are two non-process related cases that typically introduce nonlinearities to the traditional MILP representations. The simplest one is when cyclic schedules with infinite horizon are considered[245–249]. In this case, the objective function is divided by the length of the cycle time, yielding a linear fractional objective function. The rest of the model is represented using a traditional MILP representation. Specialized algorithms have been developed to address this problem[250, 251]. The second case is when a cyclic scheduling models includes average inventory equations[246]. The additional constraints in this case are nonconvex, and may require the use of a global optimization method for its solution[247, 248, 252].

In terms of process-related nonlinearities, blending equations are probably most common type that appear in scheduling models. These constraints make use of bilinear terms which are nonconvex in general, and lead to multiple local solutions. In order to solve this type of scheduling models to global optimality, rigorous global optimization techniques are used [253, 254].

One of the most complex nonconvexites in scheduling, arises in the modelling of dynamic processes. The dynamic, discrete and continuous behaviour of these systems give rise to mixed integer dynamic optimization (MIDO). This type of problems typically occur in the scheduling and control of polymerization reactors[255–259]. The the most common approach for solving MIDO problems is to discretize the system of differential equations through orthogonal collocation, yielding large scale MINLP models. Biegler[260] provides an overview of solution strategies for dynamic optimization.

7.3 Process control

Many processes in chemical engineering require the modelling of nonlinear systems, specially when representing dynamic behaviour. Additionally, discontinuity is expected in the operation and control of a process[261]. The dynamic models with mixed discrete and continuous variables are called hybrid systems (which are MIDO problems in general). Start-up and shut-down operations[262], batch systems[263], and grade transitions [258, 259] are some of the main applications of optimization of hybrid systems. Several hybrid formulations have been developed and implemented in recent years [264–271]. A comprehensive review on modelling, simulation, sensitivity analysis and optimization of hybrid systems is provided by Barton and Lee[272]. Morari and Baric[273] present a review paper on developments in the control of hybrid systems.

GDP has also been relevant in the development of models and solution methods in hybrid systems. Oldenburg and Marquardt[270] developed a GDP model of “fixed alternative sequences”, formalizing modelling

approaches proposed by several authors[272, 274]. The authors solve the GDP model using a modified version of the logic based Outer Approximation, showing the efficiency of the algorithm in finding a global solutions fast. Ruiz-Femenia et al.[80] present a logic based Outer Approximation algorithm for solving optimal control problems.

7.4 Molecular computing

A collection of works on computer aided design is provided by Achenie et al.[4]. The review paper on advances in global optimization by Floudas and Gounaris[67] provides several applications of global optimization in molecular computing.

From a mathematical programming perspective, one of the first works on molecular computing was developed by Macchietto et al.[275, 276]. The MINLP they proposed was based om group contribution accounting for the presence or absence of a group in a molecule. Churi and Achenie[277–279] further refined this model by incorporating some information on how the groups are connected to each other in the molecule. Naser and Fournier [280] proposed an alternative MINLP representation, and developed a heuristic method for solving the problem. All of this MINLP models and methods are solved using local tools, so they do not guarantee the global solution. Joback and Stephanopoulos[281, 282] proposed the use of interval arithmetic techniques, providing bounds on the properties of aggregate molecules, overcoming the multiple local optima issue.

Several applications of MINLP have been used in molecular design. Some of them include design of solvents[275, 277, 283–289], refrigerants[278, 279, 290–292], polymers[293–297], and pharmaceutical products[298, 299].

8 Concluding remarks

This work presents MINLP and GDP solution methods and applications in PSE. It is clear that MINLP has had an important impact in PSE applications. Also, the use of GDP for modelling of problems has provided a systematic method for deriving MINLP formulations. It can also be seen from this work that the use of MINLP and GDP models in PSE has increased in recent years.

The increased generation of MINLP and GDP models has also generated interest in the research of better solution methods and techniques. Improving solution methods for MINLP and GDP is a major area in the active work in operations research.

Author Information

Corresponding Author

*E-mail: grossmann@cmu.edu.

Acknowledgments

The authors would like to acknowledge financial support from the Center for Advanced Process Decision-making (CAPD).

References

[1] Grossmann, I. E.; Caballero, J. A.; Yeomans, H. Advances in mathematical programming for the synthesis of process systems. *Latin American Applied Research* **2000**, *30*, 263–284.

[2] Pinedo, M. *Scheduling: theory, algorithms, and systems*; Springer, 2012.

[3] Bertsekas, D. P.; Bertsekas, D. P.; Bertsekas, D. P.; Bertsekas, D. P. *Dynamic programming and optimal control*; Athena Scientific Belmont, 1995; Vol. 1 and 2.

[4] Achenie, L.; Venkatasubramanian, V.; Gani, R. *Computer aided molecular design: Theory and practice*; Elsevier, 2002; Vol. 12.

[5] Dakin, R. J. A tree-search algorithm for mixed programming problems. *The Computer Journal* **1965**, *8*, 250–255.

[6] Schrijver, A. *Theory of Linear and Integer Programming*; Wiley, 1986.

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

[8] Vanderbei, R. J. LOQO: An Interior Point Code for Quadratic Programming.; tech. rep.; Optimization Methods and Software, 1998.

[9] Bixby, R. E.; Rothberg, E. Progress in computational mixed integer programming A look back from the other side of the tipping point. *Annals of Operations Research* **2007**, *149*, 37–41.

[10] IBM IBM ILOG CPLEX Optimization Studio CPLEX Users Manual.; tech. rep., 2013.

[11] Optimization, G. Gurobi optimizer reference manual.; tech. rep., 2013.

[12] FICO XPRESS-MP Reference Manual.; tech. rep., 2009.

[13] Linderoth, J. T.; Ralphs, T. K. Noncommercial Software for Mixed-Integer Linear Programming. *CRC Press Operations Research Series* **2005**, 253–303.

[14] Belotti, P.; Kirches, C.; Leyffer, S.; Linderoth, J.; Luedtke, J.; Mahajan, A. Mixed-integer nonlinear optimization. *Acta Numerica* **2013**, *22*, 1–131.

[15] Bonami, P.; Kilinc, M.; Linderoth, J. T. Algorithms and Software for Solving Convex Mixed Integer Nonlinear Programs. *Chapter in Jon Lee et Sven Leyffer editors, Mixed Integer Nonlinear Programming, IMA Volumes in Mathematics and its Applications* **2012**, *154*, 1–39.

[16] Tawarmalani, M.; Sahinidis, N. V. *Convexication and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*; Kluwer Academic Publishers, 2002.

[17] Raman, R.; Grossmann, I. E. Modeling and computational techniques for logic-based integer programming. *Computers and Chemical Engineering* **1994**, *18*, 563–578.

[18] Lee, S.; Grossmann, I. E. New algorithms for nonlinear generalized disjunctive programming. *Computers and Chemical Engineering* **2000**, *24*, 2125–2141.

[19] Grossmann, I. E.; Trespalacios, F. Systematic modeling of discrete-continuous optimization models through generalized disjunctive programming. *AIChE Journal* **2013**, *59*, 3276–3295.

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

[21] Grossmann, I. E.; Ruiz, J. P. Generalized Disjunctive Programming: A Framework for Formulation and Alternative Algorithms for MINLP Optimization. *The IMA Volumes in Mathematics and its Applications* **2012**, *154*, 93–115.

[22] Balas, E. Disjunctive programming. *Annals of Discrete Mathematics* **1979**, *5*, 3–51.

[23] Clocksin, W. F.; Mellish, C. S. *Programming in Prolog*; Springer, 1981.

[24] Williams, H. P. *Model Building in Mathematical Programming*; Wiley, 1985.

[25] Biegler, L. T.; Grossmann, I. E.; Westerberg, A. W. *Systematic methods of chemical process design*; Prentice-Hall international series in the physical and chemical engineering sciences; Prentice Hall PTR, 1997.

[26] Sawaya, N. Reformulations, relaxations and cutting planes for generalized disjunctive programming. Ph.D. Thesis, Carnegie Mellon University, 2006.

[27] Grossmann, I. E.; Lee, S. Generalized convex disjunctive programming: nonlinear convex hull relaxation. *Computational Optimization and Applications* **2003**, *26*, 83–100.

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

[29] Gupta, O. K.; Ravindran, A. Branch and bound experiments in convex nonlinear integer programming. *Management Science* **1985**, *31*, 1533–1546.

[30] Nabar, S.; Schrage, L. Modeling and Solving Nonlinear Integer Programming Problems. *Presented at Annual AIChE Meeting* **1991**.

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

[32] Stubbs, R.; Mehrotra, S. A branch-and-cut method for 0-1 mixed convex programming. *Mathematical Programming* **1999**, *86*, 515–532.

[33] S., L. Integrating SQP and Branch and Bound for Mixed Integer Noninear Programming. *Computational Optimization and Applications* **2001**, *18*, 295–309.

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

[35] Fletcher, R.; Leyffer, S. Solving mixed integer nonlinear programs by Outer Approximation. *Mathematical Programming* **1994**, *66*, 327–149.

[36] Geoffrion, A. M. Generalized Benders decomposition. *Journal of Optimization Theory and Applications* **1972**, *10*, 237–260.

[37] Flippo, O. E.; Kan, A. H. G. R. Decomposition in general mathematical programming. *Mathematical Programming* **1993**, *60*, 361–382.

[38] Quesada, I.; Grossmann, I. E. An LP/NLP based branch and bound algorithm for convex MINLP optimization problems. *Computers and Chemical Engineering* **1992**, *16*, 937–947.

[39] Westerlund, T.; Pettersson, F. An extended cutting plane method for solving convex MINLP problems. *Computers and Chemical Engineering* **1995**, *19*, 131–136.

[40] Kelley, J. E. The cutting plane method for solving convex programs. *Journal of the SIAM* **1960**, *8*, 703–712.

[41] Abhishek, K.; Leyffer, S.; Linderoth, J. FilMINT: An Outer Approximation-Based Solver for Convex Mixed-Integer Nonlinear Programs. *INFORMS Journal on Computing* **2010**, *22*, 555–567.

[42] Bussieck, M. R.; Vigerske, S. MINLP solver software. *Wiley Encyclopedia of Operations Research and Management Science* **2010**.

[43] DAmbrosio, C.; Lodi, A. Mixed integer nonlinear programming tools: a practical overview. *4OR* **2011**, *9*, 329–349.

[44] Westerlund, T.; Lundqvist, K. *Alpha-ECP, Version 5.01: An interactive MINLP-solver based on the Extended Cutting Plane Method*; Åbo Akademi, 2001.

[45] Brooke, A.; Kendrick, D.; Meeraus, A.; Raman GAMS, a Users Guide. *The Scientific Press* **1998**.

[46] Bonami, P.; Biegler, L.; Conn, A.; Cornuejols, G.; Grossmann, I.; Laird, C.; Lee, J.; Lodi, A.; Margot, F.; Sawaya, N.; Wachter, A. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization* **2008**, *5*, 186–204.

[47] Wachter, A.; Biegler, L. T. On the implementation of a primal-dual interior point lter line search algorithm for large-scale nonlinear programming. *Mathematical Programming* **2006**, *106*, 25–57.

[48] Fletcher, R.; Leyffer, S. User manual for lterSQP. *University of Dundee Numerical Analysis Report NA-181*.

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

[50] Benson, H. Y.; Shanno, D. F. An exact primal-dual penalty method approach to warmstarting interior-point methods for linear programming. *Computational Optimization and Applications* **2007**, *38*, 371–399.

[51] Benson, H. Y.; Shanno, D. F. Interior-point methods for nonconvex nonlinear programming: regularization and warmstarts. *Computational Optimization and Applications* **2008**, *40*, 143–189.

[52] Leyffer, S.; tech. rep.

[53] Schweiger, C.; Floudas, C. MINOPT: A Modeling Language and Algorithmic Framework for Linear, Mixed-Integer. *Nonlinear, Dynamic, and Mixed-Integer Nonlinear Optimization* **1998**.

[54] Beaumont, N. An algorithm for disjunctive programs. *European Journal of Operational Research* **1990**, *48*, 362–371.

[55] Sawaya, N.; Grossmann, I. E. A hierarchy of relaxations for linear generalized disjunctive programming. *European Journal of Operational Research* **2012**, *216*, 70–82.

[56] Turkay, M.; Grossmann, I. E. A logic-based outer-approximation algorithm for minlp optimization of process flowsheets. *Computers and Chemical Engineering* **1996**, *20*, 959–978.

[57] Floudas, C. A. *Deterministic global optimization: theory, methods and applications*; Springer, 2000; Vol. 37.

[58] Biegler, L. T. *Nonlinear programming: concepts, algorithms, and applications to chemical processes*; SIAM, 2010.

[59] McCormick, G. P. Computability of global solutions to factorable nonconvex programs: Part IConvex underestimating problems. *Mathematical programming* **1976**, *10*, 147–175.

[60] Smith, E.; Pantelides, C. C. Global optimisation of nonconvex MINLPs. *Computers and Chemical Engineering* **1997**, *21*, S791–S796.

[61] Davis, E. Constraint propagation with interval labels. *Artificial intelligence* **1987**, *32*, 281–331.

[62] Savelsbergh, M. W. P. Preprocessing and probing techniques for mixed integer programming problems. *ORSA Journal on Computing* **1994**, *6*, 445–454.

[63] Andersen, E. D.; Andersen, K. D. Presolving in linear programming. *Mathematical Programming* **1995**, *71*, 221–245.

[64] Messine, F. Deterministic global optimization using interval constraint propagation techniques. *RAIRO-Operations Research* **2004**, *38*, 277–293.

[65] Horst, R.; Pardalos, P. M.; Romeijn, H. E. *Handbook of global optimization*; Springer, 2002; Vol. 2.

[66] Liberti, L. Introduction to global optimization.; tech. rep.; Technical report, LIX, École Polytechnique, 2008.

[67] Floudas, C. A.; Gounaris, C. E. A review of recent advances in global optimization. *Journal of Global Optimization* **2009**, *45*, 3–38.

[68] McCormick, G. P. Converting general nonlinear programming problems to separable nonlinear programming problems.; tech. rep.; DTIC Document, 1972.

[69] Tawarmalani, M.; Sahinidis, N. V. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming* **2005**, *103*, 225–249.

[70] Belotti, P.; Lee, J.; Liberti, L.; Margot, F.; Wächter, A. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods & Software* **2009**, *24*, 597–634.

[71] Belotti, P. Bound reduction using pairs of linear inequalities. *Journal of Global Optimization* **2012**, 1–33.

[72] Belotti, P.; Cafieri, S.; Lee, J.; Liberti, L. In *Combinatorial Optimization and Applications*; Springer, 2010, pp 65–76.

[73] Adjiman, C. S.; Dallwig, S.; Floudas, C. A.; Neumaier, A. A global optimization method,[alpha] BB, for general twice-differentiable constrained NLPs–I. Theoretical advances. *Computers & Chemical Engineering* **1998**, *22*, 1137–1158.

[74] Misener, R; Floudas, C. ANTIGONE: Algorithms for continuous/integer global optimization of nonlinear equations. *Submitted for publication* **2013**.

[75] Sahinidis, N. V. BARON: A general purpose global optimization software package. *Journal of Global Optimization* **1996**, *8*, 201–205.

[76] Belotti, P. Couenne: a users manual. *Department of Mathematical Sciences, Clemson University, Clemson, SC, available at http://www. coin-or. org/Couenne/couenneuser-manual. pdf, accessed April* **2009**, *23*, 2012.

[77] Lin, Y.; Schrage, L. The global solver in the LINDO API. *Optimization Methods & Software* **2009**, *24*, 657–668.

[78] Achterberg, T. SCIP: solving constraint integer programs. *Mathematical Programming Computation* **2009**, *1*, 1–41.

[79] Ruiz, J. P.; Grossmann, I. E. Strengthening of lower bounds in the global optimization of Bilinear and Concave Generalized Disjunctive Programs. *Computers & chemical engineering* **2010**, *34*, 914–930.

[80] Ruiz, J. P.; Grossmann, I. E. Using convex nonlinear relaxations in the global optimization of nonconvex generalized disjunctive programs. *Computers and Chemical Engineering* **2013**, *49*, 70–84.

[81] Balas, E. Disjunctive Programming and a Hierarchy of Relaxations for Discrete Continuous Optimization Problems. *SIAM. Journal on Algebraic and Discrete Methods* **1985**, *6*, 466–486.

[82] Ruiz, J. P.; Grossmann, I. E. A hierarchy of relaxations for nonlinear convex generalized disjunctive programming. *European Journal of Operational Research* **2012**, *218*, 38–47.

[83] Trespalacios, F.; Grossmann, I. E. Algorithmic approach for improved mixed-integer reformulations of convex Generalized Disjunctive Programs. *Informs Journal on Computing* **2014**, *Submitted for publication*.

[84] Barttfeld, M.; Aguirre, P. A.; Grossmann, I. E. Alternative representations and formulations for the economic optimization of multicomponent distillation columns. *Computers and Chemical Engineering* **2003**, *27*, 363–383.

[85] Grossmann, I. E. Mixed-integer programming approach for the synthesis of integrated process flowsheets. *Computers & chemical engineering* **1985**, *9*, 463–482.

[86] Grossmann, I. E. et al. MINLP optimization strategies and algorithms for process synthesis. **1989**.

[87] Grossmann, I. E. Mixed-integer nonlinear programming techniques for the synthesis of engineering systems. *Research in Engineering Design* **1990**, *1*, 205–228.

[88] Friedler, F; Tarjan, K; Huang, Y.; Fan, L. Graph-theoretic approach to process synthesis: axioms and theorems. *Chemical Engineering Science* **1992**, *47*, 1973–1988.

[89] Friedler, F; Tarjan, K; Huang, Y.; Fan, L. Graph-theoretic approach to process synthesis: polynomial algorithm for maximal structure generation. *Computers and Chemical Engineering* **1993**, *17*, 929–942.

[90] Smith, E. M. B.; Pantelides, C. C. Design of reaction/separation networks using detailed models. *Computers and chemical engineering* **1995**, *19*, 83–88.

[91] Papalexandri, K. P.; Pistikopoulos, E. N. Generalized modular representation framework for process synthesis. *AIChE Journal* **1996**, *42*, 1010–1032.

[92] Raman, R.; Grossmann, I. E. Symbolic integration of logic in mixed-integer linear programming techniques for process synthesis. *Computers and chemical engineering* **1993**, *17*, 909–927.

[93] Türkay, M.; Grossmann, I. E. Disjunctive programming techniques for the optimization of process systems with discontinuous investment costs-multiple size regions. *Industrial and engineering chemistry research* **1996**, *35*, 2611–2623.

[94] Türkay, M.; Grossmann, I. E. Logic-based MINLP algorithms for the optimal synthesis of process networks. *Computers and Chemical Engineering* **1996**, *20*, 959–978.

[95] Yeomans, H.; Grossmann, I. E. A systematic modeling framework of superstructure optimization in process synthesis. *Computers and Chemical Engineering* **1999**, *23*, 709–731.

[96] Vecchietti, A.; Grossmann, I. E. LOGMIP: a disjunctive 0-1 non-linear optimizer for process system models. *Computers and Chemical Engineering* **1994**, *23*, 555–565.

[97] Kocis, G. R.; Grossmann, I. E. A modelling and decomposition strategy for the MINLP optimization of process flowsheets. *Computers and chemical engineering* **1989**, *13*, 797–819.

[98] Kravanja, Z.; Grossmann, I. E. Multilevel-hierarchical MINLP synthesis of process flowsheets. *Computers and chemical engineering* **1997**, *21*, S421–S426.

[99] Daichendt, M. M.; Grossmann, I. E. Integration of hierarchical decomposition and mathematical programming for the synthesis of process flowsheets. *Computers and chemical engineering* **1998**, *22*, 147–175.

[100] Bedenik, N. I.; Pahor, B.; Kravanja, Z. An integrated strategy for the hierarchical multilevel MINLP synthesis of overall process flowsheets using the combined synthesis/analysis approach. *Computers and chemical engineering* **2004**, *28*, 693–706.

[101] Won, K. S.; Ray, T. A framework for design optimization using surrogates. *Engineering optimization* **2005**, *37*, 685–703.

[102] Shao, T.; Krishnamurty, S.; Wilmes, G. Preference-based surrogate Modeling in engineering design. *AIAA journal* **2007**, *45*, 2688–2701.

[103] Xiong, Y.; Chen, W.; Apley, D.; Ding, X. A non-stationary covariance-based Kriging method for metamodelling in engineering design. *International Journal for Numerical Methods in Engineering* **2007**, *71*, 733–756.

[104] Henao, C. A.; Maravelias, C. T. Surrogate-based superstructure optimization framework. *AIChE Journal* **2011**, *57*, 1216–1232.

[105] Kamm, B; Kamm, M In *White Biotechnology*; Springer, 2007, pp 175–204.

[106] Baliban, R. C.; Elia, J. A.; Floudas, C. A. Toward novel hybrid biomass, coal, and natural gas processes for satisfying current transportation fuel demands, 1: Process alternatives, gasification modeling, process simulation, and economic analysis. *Industrial and Engineering Chemistry Research* **2010**, *49*, 7343–7370.

[107] Martín, M.; Grossmann, I. E. Energy optimization of bioethanol production via gasification of switchgrass. *AIChE Journal* **2011**, *57*, 3408–3428.

[108] Martín, M.; Grossmann, I. E. Process optimization of FT-diesel production from lignocellulosic switchgrass. *Industrial and Engineering Chemistry Research* **2011**, *50*, 13485–13499.

[109] Santibanez-Aguilar, J. E.; González-Campos, J. B.; Ponce-Ortega, J. M.; Serna-González, M.; El-Halwagi, M. M. Optimal planning of a biomass conversion system considering economic and environmental aspects. *Industrial and Engineering Chemistry Research* **2011**, *50*, 8558–8570.

[110] Murat Sen, S; Henao, C. A.; Braden, D. J.; Dumesic, J. A.; Maravelias, C. T. Catalytic conversion of lignocellulosic biomass to fuels: Process development and technoeconomic evaluation. *Chemical Engineering Science* **2012**, *67*, 57–67.

[111] Tay, D. H.; Kheireddine, H.; Ng, D. K.; El-Halwagi, M. M.; Tan, R. R. Conceptual synthesis of gasification-based biorefineries using thermodynamic equilibrium optimization models. *Industrial and Engineering Chemistry Research* **2011**, *50*, 10681–10695.

[112] Tay, D. H.; Ng, D. K.; Sammons Jr, N. E.; Eden, M. R. Fuzzy optimization approach for the synthesis of a sustainable integrated biorefinery. *Industrial and Engineering Chemistry Research* **2011**, *50*, 1652–1665.

[113] Kim, J.; Sen, S. M.; Maravelias, C. T. An optimization-based assessment framework for biomass-to-fuel conversion strategies. *Energy and Environmental Science* **2013**, *6*, 1093–1104.

[114] Wang, B.; Gebreslassie, B. H.; You, F. Sustainable design and synthesis of hydrocarbon biorefinery via gasification pathway: Integrated life cycle assessment and technoeconomic analysis with multi-objective superstructure optimization. *Computers and Chemical Engineering* **2013**, *52*, 55–76.

[115] Liu, P.; Pistikopoulos, E. N.; Li, Z. A multi-objective optimization approach to polygeneration energy systems design. *AIChE journal* **2010**, *56*, 1218–1234.

[116] Liu, P.; Georgiadis, M. C.; Pistikopoulos, E. N. Advances in energy systems engineering. *Industrial and Engineering Chemistry Research* **2010**, *50*, 4915–4926.

[117] Chen, Y.; Adams, T. A.; Barton, P. I. Optimal design and operation of static energy polygeneration systems. *Industrial and Engineering Chemistry Research* **2010**, *50*, 5099–5113.

[118] Ellepola, J.; Thijssen, N.; Grievink, J.; Baak, G.; Avhale, A.; Schijndel, J. van Development of a synthesis tool for Gas-To-Liquid complexes. *Computers and Chemical Engineering* **2012**, *42*, 2–14.

[119] Hildebrandt, D.; Biegler, L. T. Synthesis of chemical reactor networks.; tech. rep.; Carnegie Mellon University, 1995.

[120] Achenie, L. E.; Biegler, L. T. Algorithmic synthesis of chemical reactor networks using mathematical programming. *Industrial and engineering chemistry fundamentals* **1986**, *25*, 621–627.

[121] Achenie, L.; Biegler, L. A superstructure based approach to chemical reactor network synthesis. *Computers and chemical engineering* **1990**, *14*, 23–40.

[122] Kokossis, A. C.; Christodoulos A, F. Optimization of complex reactor networksI. Isothermal operation. *Chemical Engineering Science* **1990**, *45*, 595–614.

[123] Kokossis, A. C.; Floudas, C. A. Synthesis of isothermal reactorseparatorrecycle systems. *Chemical engineering science* **1991**, *46*, 1361–1383.

[124] Kokossis, A. C.; Floudas, C. Optimization of complex reactor networksII. Nonisothermal operation. *Chemical Engineering Science* **1994**, *49*, 1037–1051.

[125] Esposito, W.; Floudas, C. Deterministic global optimization in isothermal reactor network synthesis. *Journal of Global Optimization* **2002**, *22*, 59–95.

[126] Horn, F In *Third European symposium on chemical reaction engineering*, 1964, pp 1–10.

[127] Glasser, D.; Crowe, C.; Hildebrandt, D. A geometric approach to steady flow reactors: the attainable region and optimization in concentration space. *Industrial and engineering chemistry research* **1987**, *26*, 1803–1810.

[128] Hildebrandt, D; Glasser, D The attainable region and optimal reactor structures. *Chemical Engineering Science* **1990**, *45*, 2161–2168.

[129] Hildebrandt, D.; Glasser, D.; Crowe, C. M. Geometry of the attainable region generated by reaction and mixing: with and without constraints. *Industrial and engineering chemistry research* **1990**, *29*, 49–58.

[130] Feinberg, M.; Hildebrandt, D. Optimal reactor design from a geometric viewpointI. Universal properties of the attainable region. *Chemical Engineering Science* **1997**, *52*, 1637–1665.

[131] Balakrishna, S.; Biegler, L. T. Constructive targeting approaches for the synthesis of chemical reactor networks. *Industrial and engineering chemistry research* **1992**, *31*, 300–312.

[132] Balakrishna, S.; Biegler, L. T. Targeting strategies for the synthesis and energy integration of non-isothermal reactor networks. *Industrial and engineering chemistry research* **1992**, *31*, 2152–2164.

[133] Balakrishna, S.; Biegler, L. T. A unified approach for the simultaneous synthesis of reaction, energy, and separation systems. *Industrial & engineering chemistry research* **1993**, *32*, 1372–1382.

[134] Lakshmanan, A.; Biegler, L. T. Synthesis of optimal chemical reactor networks. *Industrial and engineering chemistry research* **1996**, *35*, 1344–1353.

[135] Rooney, W. C.; Hausberger, B. P.; Biegler, L. T.; Glasser, D. Convex attainable region projections for reactor network synthesis. *Computers and Chemical Engineering* **2000**, *24*, 225–229.

[136] Sargent, R.; Gaminibandara, K Optimum design of plate distillation columns.

[137] Sargent, R. A functional approach to process synthesis and its application to distillation systems. *Computers and chemical engineering* **1998**, *22*, 31–45.

[138] Viswanathan, J.; Grossmann, I. E. Optimal feed locations and number of trays for distillation columns with multiple feeds. *Industrial and engineering chemistry research* **1993**, *32*, 2942–2949.

[139] Bauer, M.; Stichlmair, J Design and economic optimization of azeotropic distillation processes using mixed-integer nonlinear programming. *Computers and chemical engineering* **1998**, *22*, 1271–1286.

[140] Dünnebier, G.; Pantelides, C. C. Optimal design of thermally coupled distillation columns. *Industrial and engineering chemistry research* **1999**, *38*, 162–176.

[141] Yeomans, H.; Grossmann, I. E. Disjunctive programming models for the optimal design of distillation columns and separation sequences. *Industrial and Engineering Chemistry Research* **2000**, *39*, 1637–1648.

[142] Rudd, D. F.; Watson, C. C. *Strategy of process engineering*; Wiley New York, New York, 1968.

[143] Caballero, J. A.; Grossmann, I. E. Aggregated models for integrated distillation systems. *Industrial and engineering chemistry research* **1999**, *38*, 2330–2344.

[144] Andrecovich, M.; Westerberg, A. An MILP formulation for heat-integrated distillation sequence synthesis. *AIChE journal* **1985**, *31*, 1461–1474.

[145] Novak, Z; Kravanja, Z; Grossmann, I. Simultaneous synthesis of distillation sequences in overall process schemes using an improved MINLP approach. *Computers and chemical engineering* **1996**, *20*, 1425–1440.

[146] Paules, G.; Floudas, C. Stochastic programming in process synthesis: a two-stage model with MINLP recourse for multiperiod heat-integrated distillation sequences. *Computers and chemical engineering* **1992**, *16*, 189–210.

[147] Floudas, C.; Paules IV, G. A mixed-integer nonlinear programming formulation for the synthesis of heat-integrated distillation sequences. *Computers and chemical engineering* **1988**, *12*, 531–546.

[148] Kaibel, G. Distillation columns with vertical partitions. *Chemical engineering and technology* **1987**, *10*, 92–98.

[149] Agrawal, R.; Fidkowski, Z. T. More operable arrangements of fully thermally coupled distillation columns. *AIChE Journal* **1998**, *44*, 2565–2568.

[150] Agrawal, R A method to draw fully thermally coupled distillation column configurations for multicomponent distillation. *Chemical Engineering Research and Design* **2000**, *78*, 454–464.

[151] Caballero, J. A.; Grossmann, I. E. Thermodynamically equivalent configurations for thermally coupled distillation. *AIChE journal* **2003**, *49*, 2864–2884.

[152] Serra, M.; Espuna, A.; Puigjaner, L. Control and optimization of the divided wall column. *Chemical Engineering and Processing: Process Intensification* **1999**, *38*, 549–562.

[153] Serra, M.; Perrier, M.; Espuna, A.; Puigjaner, L. Study of the divided wall column controllability: influence of design and operation. *Computers and Chemical Engineering* **2000**, *24*, 901–907.

[154] Serra, M; Perrier, M; Espuna, A; Puigjaner, L Analysis of different control possibilities for the divided wall column: feedback diagonal and dynamic matrix control. *Computers and Chemical Engineering* **2001**, *25*, 859–866.

[155] Serra, M.; Espuna, A.; Puigjaner, L. Controllability of different multicomponent distillation arrangements. *Industrial and engineering chemistry research* **2003**, *42*, 1773–1782.

[156] Wolff, E. A.; Skogestad, S. Operation of integrated three-product (Petlyuk) distillation columns. *Industrial and engineering chemistry research* **1995**, *34*, 2094–2103.

[157] Agrawal, R. Synthesis of multicomponent distillation column configurations. *AIChE journal* **2003**, *49*, 379–401.

[158] Caballero, J. A.; Grossmann, I. E. Structural considerations and modeling in the synthesis of heat-integrated-thermally coupled distillation sequences. *Industrial and engineering chemistry research* **2006**, *45*, 8454–8474.

[159] Giridhar, A.; Agrawal, R. Synthesis of distillation configurations: I. Characteristics of a good search space. *Computers and chemical engineering* **2010**, *34*, 73–83.

[160] Giridhar, A.; Agrawal, R. Synthesis of distillation configurations. II: A search formulation for basic configurations. *Computers and chemical engineering* **2010**, *34*, 84–95.

[161] Ivakpour, J.; Kasiri, N. Synthesis of distillation column sequences for nonsharp separations. *Industrial and Engineering Chemistry Research* **2009**, *48*, 8635–8649.

[162] Shah, V. H.; Agrawal, R. A matrix method for multicomponent distillation sequences. *AIChE Journal* **2010**, *56*, 1759–1775.

[163] Caballero, J. A.; Grossmann, I. E. Generalized disjunctive programming model for the optimal synthesis of thermally linked distillation columns. *Industrial and engineering chemistry research* **2001**, *40*, 2260–2274.

[164] Caballero, J. A.; Grossmann, I. E. Design of distillation sequences: from conventional to fully thermally coupled distillation systems. *Computers and chemical engineering* **2004**, *28*, 2307–2329.

[165] Furman, K. C.; Sahinidis, N. V. A critical review and annotated bibliography for heat exchanger network synthesis in the 20th century. *Industrial and Engineering Chemistry Research* **2002**, *41*, 2335–2370.

[166] Morar, M.; Agachi, P. S. Review: Important contributions in development and improvement of the heat integration techniques. *Computers and chemical engineering* **2010**, *34*, 1171–1179.

[167] Cerda, J.; Westerberg, A. W.; Mason, D.; Linnhoff, B. Minimum utility usage in heat exchanger network synthesis a transportation problem. *Chemical Engineering Science* **1983**, *38*, 373–387.

[168] Papoulias, S. A.; Grossmann, I. E. A structural optimization approach in process synthesisII: Heat recovery networks. *Computers and Chemical Engineering* **1983**, *7*, 707–721.

[169] Jezowski, J.; Friedler, F. A simple approach for maximum heat recovery calculations. *Chemical engineering science* **1992**, *47*, 1481–1494.

[170] Galli, M.; Cerda, J A designer-controlled framework for the synthesis of heat exchanger networks involving non-isothermal mixers and multiple units over split streams. *Computers and chemical engineering* **1998**, *22*, S813–S816.

[171] Galli, M.; Cerdá, J., et al. Synthesis of structural-constrained heat exchanger networks–I. Series networks. *Computers and chemical engineering* **1998**, *22*, 819–839.

[172] Galli, M.; Cerdá, J., et al. Synthesis of structural-constrained heat exchanger networks-II Split Networks. *Computers and chemical engineering* **1998**, *22*, 1017–1035.

[173] Gundersen, T.; Grossmann, I. E. Improved optimization strategies for automated heat exchanger network synthesis through physical insights. *Computers and chemical engineering* **1990**, *14*, 925–944.

[174] Gundersen, T.; Duvold, S.; Hashemi-Ahmady, A. An extended vertical MILP model for heat exchanger network synthesis. *Computers and chemical engineering* **1996**, *20*, S97–S102.

[175] Floudas, C. A.; Ciric, A. R.; Grossmann, I. E. Automatic synthesis of optimum heat exchanger network configurations. *AIChE Journal* **1986**, *32*, 276–290.

[176] Yuan, X; Pibouleau, L; Domenech, S Experiments in process synthesis via mixed-integer programming. *Chemical Engineering and Processing: Process Intensification* **1989**, *25*, 99–116.

[177] Floudas, C. A.; Ciric, A. R. Strategies for overcoming uncertainties in heat exchanger network synthesis. *Computers and chemical engineering* **1989**, *13*, 1133–1152.

[178] Floudas, C. A.; Ciric, A. R. Corrigendum Strategies for Overcoming Uncertainties in Heat Exchanger Network Synthesis. *Computers and chemical engineering* **1990**, *14*.

[179] Ciric, A. R.; Floudas, C. A. Application of the simultaneous match-network optimization approach to the pseudo-pinch problem. *Computers and chemical engineering* **1990**, *14*, 241–250.

[180] Ciric, A. R.; Floudas, C. A. Heat exchanger network synthesis without decomposition. *Computers and Chemical Engineering* **1991**, *15*, 385–396.

[181] Yee, T. F.; Grossmann, I. E.; Kravanja, Z. Simultaneous optimization models for heat integrationI. Area and energy targeting and modeling of multi-stream exchangers. *Computers and chemical engineering* **1990**, *14*, 1151–1164.

[182] Yee, T. F.; Grossmann, I. E. Simultaneous optimization models for heat integrationII. Heat exchanger network synthesis. *Computers and Chemical Engineering* **1990**, *14*, 1165–1184.

[183] Konukman, A. E. Ş.; Çamurdan, M. C.; Akman, U. Simultaneous flexibility targeting and synthesis of minimum-utility heat-exchanger networks with superstructure-based MILP formulation. *Chemical Engineering and Processing: Process Intensification* **2002**, *41*, 501–518.

[184] Chen, C.-L.; Hung, P.-S. Simultaneous synthesis of flexible heat-exchange networks with uncertain source-stream temperatures and flow rates. *Industrial and engineering chemistry research* **2004**, *43*, 5916–5928.

[185] Zhang, W. V. et al. Design of flexible heat exchanger network for multi-period operation. *Chemical engineering science* **2006**, *61*, 7730–7753.

[186] Frausto-Hernández, S.; Rico-Ramırez, V; Jiménez-Gutiérrez, A.; Hernández-Castro, S. MINLP synthesis of heat exchanger networks considering pressure drop effects. *Computers and chemical engineering* **2003**, *27*, 1143–1152.

[187] Serna-González, M.; Ponce-Ortega, J. M.; Jiménez-Gutiérrez, A. Two-level optimization algorithm for heat exchanger networks including pressure drop considerations. *Industrial and engineering chemistry research* **2004**, *43*, 6766–6773.

[188] Ponce-Ortega, J. M.; Jiménez-Gutiérrez, A.; Grossmann, I. E. Optimal synthesis of heat exchanger networks involving isothermal process streams. *Computers and Chemical Engineering* **2008**, *32*, 1918–1942.

[189] Duran, M. A.; Grossmann, I. E. Simultaneous optimization and heat integration of chemical processes. *AIChE Journal* **1986**, *32*, 123–138.

[190] Grossmann, I. E.; Yeomans, H.; Kravanja, Z. A rigorous disjunctive optimization model for simultaneous flowsheet optimization and heat integration. *Computers and chemical engineering* **1998**, *22*, S157–S164.

[191] Navarro-Amorós, M. A.; Caballero, J. A.; Ruiz-Femenia, R.; Grossmann, I. E. An alternative disjunctive optimization model for heat integration with variable temperatures. *Computers and Chemical Engineering* **2013**.

[192] Petroulas, T; Reklaitis, G. Computer-aided synthesis and design of plant utility systems. *AIChE journal* **1984**, *30*, 69–78.

[193] Papoulias, S. A.; Grossmann, I. E. A structural optimization approach in process synthesisI: utility systems. *Computers and Chemical Engineering* **1983**, *7*, 695–706.

[194] Bruno, J.; Fernandez, F; Castells, F; Grossmann, I. A rigorous MINLP model for the optimal synthesis and operation of utility plants. *Chemical Engineering Research and Design* **1998**, *76*, 246–258.

[195] Savola, T.; Tveit, T.-M.; Fogelholm, C.-J. A MINLP model including the pressure levels and multiperiods for CHP process optimisation. *Applied thermal engineering* **2007**, *27*, 1857–1867.

[196] Jezowski, J. Review of water network design methods with literature annotations. *Industrial and Engineering Chemistry Research* **2010**, *49*, 4475–4516.

[197] Wang, Y. P.; Smith, R. Wastewater minimisation. *Chemical Engineering Science* **1994**, *49*, 981–1006.

[198] Wang, Y.-P.; Smith, R. Design of distributed effluent treatment systems. *Chemical Engineering Science* **1994**, *49*, 3127–3145.

[199] Bagajewicz, M. J.; Faria, D. C. On the appropriate architecture of the water/wastewater allocation problem in process plants. *Computer aided chemical engineering* **2009**, *26*, 1–20.

[200] Meyer, C. A.; Floudas, C. A. Global optimization of a combinatorially complex generalized pooling problem. *AIChE journal* **2006**, *52*, 1027–1037.

[201] Papalexandri, K.; Pistikopoulos, E.; Floudas, A Mass-exchange networks for waste minimization-a simultaneous approach. *Chemical engineering research and design* **1994**, *72*, 279–294.

[202] Szitkai, Z; Farkas, T; Kravanja, Z.; Lelkes, Z; Rev, E; Fonyo, Z A new MINLP model for mass exchange network synthesis. *Computer Aided Chemical Engineering* **2003**, *14*, 323–328.

[203] Bagajewicz, M.; Savelski, M. On the use of linear models for the design of water utilization systems in process plants with a single contaminant. *Chemical engineering research and design* **2001**, *79*, 600–610.

[204] Savelski, M. J.; Bagajewicz, M. J. On the optimality conditions of water utilization systems in process plants with single contaminants. *Chemical Engineering Science* **2000**, *55*, 5035–5048.

[205] Bagajewicz, M. J.; Rivas, M.; Savelski, M. J. A robust method to obtain optimal and sub-optimal design and retrofit solutions of water utilization systems with multiple contaminants in process plants. *Computers and Chemical Engineering* **2000**, *24*, 1461–1466.

[206] Savelski, M.; Bagajewicz, M. On the necessary conditions of optimality of water utilization systems in process plants with multiple contaminants. *Chemical engineering science* **2003**, *58*, 5349–5362.

[207] Koppol, A. P.; Bagajewicz, M. J.; Dericks, B. J.; Savelski, M. J. On zero water discharge solutions in the process industry. *Advances in Environmental Research* **2004**, *8*, 151–171.

[208] Wałczyk, K.; Jeżowski, J. A single stage approach for designing water networks with multiple contaminants. *Computer Aided Chemical Engineering* **2008**, *25*, 719–724.

[209] Lovelady, E. M.; El-Halwagi, M.; Krishnagopalan, G. A. An integrated approach to the optimisation of water usage and discharge in pulp and paper plants. *International journal of environment and pollution* **2007**, *29*, 274–307.

[210] Li, B.-H.; Chang, C.-T. A simple and efficient initialization strategy for optimizing water-using network designs. *Industrial & Engineering Chemistry Research* **2007**, *46*, 8781–8786.

[211] Teles, J.; Castro, P. M.; Novais, A. Q. LP-based solution strategies for the optimal design of industrial water networks with multiple contaminants. *Chemical Engineering Science* **2008**, *63*, 376–394.

[212] Takama, N; Kuriyama, T; Shiroko, K; Umeda, T Optimal water allocation in a petroleum refinery. *Computers and Chemical Engineering* **1980**, *4*, 251–258.

[213] Doyle, S.; Smith, R Targeting water reuse with multiple contaminants. *Process safety and environmental protection* **1997**, *75*, 181–189.

[214] Ullmer, C; Kunde, N; Lassahn, A; Gruhn, G; Schulz, K WADO: water design optimizationmethodology and software for the synthesis of process water systems. *Journal of Cleaner Production* **2005**, *13*, 485–494.

[215] Gunaratnam, M; Alva-Argaez, A; Kokossis, A; Kim, J.-K.; Smith, R Automated design of total water systems. *Industrial and engineering chemistry research* **2005**, *44*, 588–599.

[216] Hernandez-Suarez, R.; Castellanos-Fernandez, J.; Zamora, J. M. Superstructure decomposition and parametric optimization approach for the synthesis of distributed wastewater treatment networks. *Industrial and engineering chemistry research* **2004**, *43*, 2175–2191.

[217] Tsai, M.-J.; Chang, C.-T. Water usage and treatment network design using genetic algorithms. *Industrial and engineering chemistry research* **2001**, *40*, 4874–4888.

[218] Jeżowski, J.; Bochenek, R.; Poplewski, G. On application of stochastic optimization techniques to designing heat exchanger-and water networks. *Chemical Engineering and Processing: Process Intensification* **2007**, *46*, 1160–1174.

[219] Lavric, V.; Iancu, P.; Pleşu, V. Genetic algorithm optimisation of water consumption and wastewater network topology. *Journal of Cleaner Production* **2005**, *13*, 1405–1415.

[220] Zamora, J. M.; Grossmann, I. E. Continuous global optimization of structured process systems models. *Computers and Chemical Engineering* **1998**, *22*, 1749–1770.

[221] Karuppiah, R.; Grossmann, I. E. Global optimization of multiscenario mixed integer nonlinear programming models arising in the synthesis of integrated water networks under uncertainty. *Computers and Chemical Engineering* **2008**, *32*, 145–160.

[222] Ahmetović, E.; Grossmann, I. E. Global superstructure optimization for the design of integrated process water networks. *AIChE journal* **2011**, *57*, 434–457.

[223] Dong, H.-G.; Lin, C.-Y.; Chang, C.-T. Simultaneous optimization approach for integrated water-allocation and heat-exchange networks. *Chemical Engineering Science* **2008**, *63*, 3664–3678.

[224] Dong, H.-G.; Lin, C.-Y.; Chang, C.-T. Simultaneous optimization strategy for synthesizing heat exchanger networks with multi-stream mixers. *chemical engineering research and design* **2008**, *86*, 299–309.

[225] Zhou, R.-J.; Li, L.-J.; Xiao, W.; Dong, H.-G. Simultaneous optimization of batch process schedules and water-allocation network. *Computers and Chemical Engineering* **2009**, *33*, 1153–1168.

[226] Li, L.-J.; Zhou, R.-J.; Dong, H.-G. State-time-space superstructure-based MINLP formulation for batch water-allocation network design. *Industrial and Engineering Chemistry Research* **2009**, *49*, 236–251.

[227] Li, L.-J.; Zhou, R.-J.; Dong, H.-G.; Grossmann, I. E. Separation network design with mass and energy separating agents. *Computers and Chemical Engineering* **2011**, *35*, 2005–2016.

[228] Bagajewicz, M. J.; Manousiouthakis, V. Mass/heat-exchange network representation of distillation networks. *AIChE Journal* **1992**, *38*, 1769–1800.

[229] Bagajewicz, M. J.; Pham, R.; Manousiouthakis, V. On the state space approach to mass/heat exchanger network design. *Chemical Engineering Science* **1998**, *53*, 2595–2621.

[230] Pochet, Y.; Wolsey, L. A. *Production planning by mixed integer programming*; Springer, 2006.

[231] Kallrath, J. Planning and scheduling in the process industry. *OR spectrum* **2002**, *24*, 219–250.

[232] Harjunkoski, I.; Maravelias, C.; Bongers, P.; Castro, P.; Engell, S.; Grossmann, I.; Hooker, J.; Mendez, C.; Sand, G.; Wassick, J. Scope for Industrial Applications of Production Scheduling Models and Solution Methods. *Computers and Chemical Engineering* **2014**, *Submitted for publication*.

[233] Maravelias, C. T.; Sung, C. Integration of production planning and scheduling: Overview, challenges and opportunities. *Computers and Chemical Engineering* **2009**, *33*, 1919–1930.

[234] Moro, L. F. L.; Zanin, A. C.; Pinto, J. M. A planning model for refinery diesel production. *Computers and Chemical Engineering* **1998**, *22*, S1039–S1042.

[235] Pinto, J. M.; Moro, L. F. L. A planning model for petroleum refineries. *Brazilian Journal of Chemical Engineering* **2000**, *17*, 575–586.

[236] Zhang, J; Zhu, X.; Towler, G. A simultaneous optimization strategy for overall integration in refinery planning. *Industrial and engineering chemistry research* **2001**, *40*, 2640–2653.

[237] Elkamel, A; Ba-Shammakh, M; Douglas, P; Croiset, E An optimization approach for integrating planning and CO2 emission reduction in the petroleum refining industry. *Industrial and Engineering Chemistry Research* **2008**, *47*, 760–776.

[238] Neiro, S. M.; Pinto, J. M. Multiperiod optimization for production planning of petroleum refineries. *Chem. Eng. Comm.* **2005**, *192*, 62–88.

[239] Alattas, A. M.; Grossmann, I. E.; Palou-Rivera, I. Refinery Production Planning: Multiperiod MINLP with Nonlinear CDU Model. *Industrial and Engineering Chemistry Research* **2012**, *51*, 12852–12861.

[240] Geddes, R. A general index of fractional distillation power for hydrocarbon mixtures. *AIChE Journal* **1958**, *4*, 389–392.

[241] Floudas, C. A.; Lin, X. Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. *Computers and Chemical Engineering* **2004**, *28*, 2109–2129.

[242] Méndez, C. A.; Cerdá, J.; Grossmann, I. E.; Harjunkoski, I.; Fahl, M. State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Computers and Chemical Engineering* **2006**, *30*, 913–946.

[243] Méndez, C. A.; Grossmann, I. E.; Harjunkoski, I.; Kaboré, P. A simultaneous optimization approach for off-line blending and scheduling of oil-refinery operations. *Computers and chemical engineering* **2006**, *30*, 614–634.

[244] Castro, P. M.; Grossmann, I. E. Generalized disjunctive programming as a systematic modeling framework to derive scheduling formulations. *Industrial and Engineering Chemistry Research* **2012**, *51*, 5781–5792.

[245] Castro, P. M.; Barbosa-Póvoa, A. P.; Matos, H. A. Optimal periodic scheduling of batch plants using RTN-based discrete and continuous-time formulations: a case study approach. *Industrial and engineering chemistry research* **2003**, *42*, 3346–3360.

[246] Pinto, J.; Grossmann, I. Optimal cyclic scheduling of multistage continuous multiproduct plants. *Computers and Chemical Engineering* **1994**, *18*, 797–816.

[247] Sahinidis, N. V.; Grossmann, I. E. Reformulation of multiperiod MILP models for planning and scheduling of chemical processes. *Computers and Chemical Engineering* **1991**, *15*, 255–272.

[248] Sahinidis, N.; Grossmann, I. E. MINLP model for cyclic multiproduct scheduling on continuous parallel lines. *Computers and chemical engineering* **1991**, *15*, 85–103.

[249] Shah, N; Pantelides, C. C.; Sargent, R. Optimal periodic scheduling of multipurpose batch plants. *Annals of Operations Research* **1993**, *42*, 193–228.

[250] Pochet, Y.; Warichet, F. A tighter continuous time formulation for the cyclic scheduling of a mixed plant. *Computers and Chemical Engineering* **2008**, *32*, 2723–2744.

[251] You, F.; Castro, P. M.; Grossmann, I. E. Dinkelbach's algorithm as an efficient method to solve a class of MINLP models for large-scale cyclic scheduling problems. *Computers and Chemical Engineering* **2009**, *33*, 1879–1889.

[252] Horst, R.; Tuy, H. *Global optimization: Deterministic approaches*; Springer, 1996.

[253] Misener, R.; Floudas, C. A. Global optimization of large-scale generalized pooling problems: quadratically constrained MINLP models. *Industrial and Engineering Chemistry Research* **2010**, *49*, 5424–5438.

[254] Kolodziej, S. P.; Grossmann, I. E.; Furman, K. C.; Sawaya, N. W. A discretization-based approach for the optimization of the multiperiod blend scheduling problem. *Computers and Chemical Engineering* **2013**.

[255] Flores-Tlacuahuac, A.; Grossmann, I. E. Simultaneous cyclic scheduling and control of a multiproduct CSTR. *Industrial and engineering chemistry research* **2006**, *45*, 6698–6712.

[256] Mishra, B. V.; Mayer, E.; Raisch, J.; Kienle, A. Short-term scheduling of batch processes. A comparative study of different approaches. *Industrial and engineering chemistry research* **2005**, *44*, 4022–4034.

[257] Nyström, R. H.; Franke, R.; Harjunkoski, I.; Kroll, A. Production campaign planning including grade transition sequencing and dynamic optimization. *Computers and chemical engineering* **2005**, *29*, 2163–2179.

[258] Prata, A.; Oldenburg, J.; Kroll, A.; Marquardt, W. Integrated scheduling and dynamic optimization of grade transitions for a continuous polymerization reactor. *Computers and Chemical Engineering* **2008**, *32*, 463–476.

[259] Terrazas-Moreno, S.; Flores-Tlacuahuac, A.; Grossmann, I. E. Simultaneous cyclic scheduling and optimal control of polymerization reactors. *AIChE journal* **2007**, *53*, 2301–2315.

[260] Biegler, L. T. An overview of simultaneous strategies for dynamic optimization. *Chemical Engineering and Processing: Process Intensification* **2007**, *46*, 1043–1053.

[261] Barton, P. I.; Lee, C. K. Design of process operations using hybrid dynamic optimization. *Computers and chemical engineering* **2004**, *28*, 955–969.

[262] Raghunathan, A. U.; Soledad Diaz, M; Biegler, L. T. An MPEC formulation for dynamic optimization of distillation operations. *Computers and chemical engineering* **2004**, *28*, 2037–2052.

[263] Oldenburg, J.; Marquardt, W.; Heinz, D.; Leineweber, D. B. Mixed-logic dynamic optimization applied to batch distillation process design. *AIChE journal* **2003**, *49*, 2900–2917.

[264] Lu, J.; Liao, L.-z.; Nerode, A.; Taylor, J. H. In *Decision and Control, 1993., Proceedings of the 32nd IEEE Conference on*, 1993, pp 2292–2297.

[265] Avraam, M.; Shah, N; Pantelides, C. Modelling and optimisation of general hybrid systems in the continuous time domain. *Computers and chemical engineering* **1998**, *22*, S221–S228.

[266] Bemporad, A.; Morari, M. Control of systems integrating logic, dynamics, and constraints. *Automatica* **1999**, *35*, 407–427.

[267] Buss, M.; Stryk, O. v.; Bulirsch, R.; Schmidt, G. Towards hybrid optimal control. *at-Automatisierungstechnik* **2000**, *48*, 448–459.

[268] Stursberg, O.; Panek, S. In *Hybrid systems: Computation and control*; Springer, 2002, pp 421–435.

[269] Zabiri, H.; Samyudia, Y. A hybrid formulation and design of model predictive control for systems under actuator saturation and backlash. *Journal of Process Control* **2006**, *16*, 693–709.

[270] Oldenburg, J.; Marquardt, W. Disjunctive modeling for optimal control of hybrid systems. *Computers and Chemical Engineering* **2008**, *32*, 2346–2364.

[271] Moro, L. F. L.; Grossmann, I. E. A Mixed-integer Model Predictive Control Formulation for Linear Systems. *Computers and Chemical Engineering* **2013**.

[272] Barton, P. I.; Lee, C. K. Modeling, simulation, sensitivity analysis, and optimization of hybrid systems. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* **2002**, *12*, 256–289.

[273] Morari, M.; Barić, M. Recent developments in the control of constrained hybrid systems. *Computers and chemical engineering* **2006**, *30*, 1619–1631.

[274] Obertopp, T; Spieker, A; Gilles, E. Optimierung hybrider Prozesse in der Verfahrenstechnik. *Oberhausener UMSICHT-Tage: Rechneranwendungen in der Verfahrenstechnik, UMSICHT-Schriftenreihe* **1998**, *7*, 5.1–5.18.

[275] Macchietto, S; Odele, O; Omatsone, O Design on optimal solvents for liquid-liquid extraction and gas absorption processes. *Chemical engineering research and design* **1990**, *68*, 429–433.

[276] Odele, O; Macchietto, S Computer aided molecular design: A novel method for optimal solvent selection. *Fluid Phase Equilibria* **1993**, *82*, 47–54.

[277] Churi, N.; Achenie, L. E. K. Novel mathematical programming model for computer aided molecular design. *Industrial and engineering chemistry research* **1996**, *35*, 3788–3794.

[278] Churi, N.; Achenie, L. E. On the Use of a Mixed Integer Non-linear Programming Model for Refrigerant Design. *International Transactions in Operational Research* **1997**, *4*, 45–54.

[279] Churi, N.; Achenie, L. E. The optimal design of refrigerant mixtures for a two-evaporator refrigeration system. *Computers and chemical engineering* **1997**, *21*, S349–S354.

[280] Naser, S.; Fournier, R. A system for the design of an optimum liquid-liquid extractant molecule. *Computers and chemical engineering* **1991**, *15*, 397–414.

[281] Joback, K. G. Designing molecules possessing desired physical property values. Ph.D. Thesis, Massachusetts Institute of Technology, 1989.

[282] Joback, K. G.; Stephanopoulos, G. Searching spaces of discrete solutions: The design of molecules possessing desired physical properties. *Advances in chemical engineering* **1995**, *21*, 257–311.

[283] Pistikopoulos, E.; Stefanis, S. Optimal solvent design for environmental impact minimization. *Computers and chemical engineering* **1998**, *22*, 717–733.

[284] Sinha, M.; Achenie, L. E.; Ostrovsky, G. M. Environmentally benign solvent design by global optimization. *Computers and Chemical Engineering* **1999**, *23*, 1381–1394.

[285] Ostrovsky, G. M.; Achenie, L. E.; Sinha, M. On the solution of mixed-integer nonlinear programming models for computer aided molecular design. *Computers and chemistry* **2002**, *26*, 645–660.

[286] Ostrovsky, G. M.; Achenie, L. E.; Sinha, M. A reduced dimension branch-and-bound algorithm for molecular design. *Computers and chemical engineering* **2003**, *27*, 551–567.

[287] Wang, Y.; Achenie, L. E. A hybrid global optimization approach for solvent design. *Computers and chemical engineering* **2002**, *26*, 1415–1425.

[288] Wang, Y.; Achenie, L. E. Computer aided solvent design for extractive fermentation. *Fluid Phase Equilibria* **2002**, *201*, 1–18.

[289] Sinha, M.; Achenie, L. E.; Gani, R. Blanket wash solvent blend design using interval analysis. *Industrial and engineering chemistry research* **2003**, *42*, 516–527.

[290] Duvedi, A. P.; Achenie, L. E. Designing environmentally safe refrigerants using mathematical programming. *Chemical Engineering Science* **1996**, *51*, 3727–3739.

[291] Sahinidis, N. V.; Tawarmalani, M. Applications of global optimization to process and molecular design. *Computers and Chemical Engineering* **2000**, *24*, 2157 –2169.

[292] Sahinidis, N. V.; Tawarmalani, M.; Yu, M. Design of alternative refrigerants via global optimization. *AIChE Journal* **2003**, *49*, 1761–1775.

[293] Maranas, C. D. Optimal computer-aided molecular design: A polymer design case study. *Industrial and engineering chemistry research* **1996**, *35*, 3403–3414.

[294] Maranas, C. D. Optimal molecular design under property prediction uncertainty. *AIChE Journal* **1997**, *43*, 1250–1264.

[295] Maranas, C. D. Optimization accounting for property prediction uncertainty in polymer design. *Computers and chemical engineering* **1997**, *21*, S1019–S1024.

[296] Vaidyanathan, R.; Gowayed, Y.; El-Halwagi, M. Computer-aided design of fiber reinforced polymer composite products. *Computers and chemical engineering* **1998**, *22*, 801–808.

[297] Camarda, K. V.; Maranas, C. D. Optimization in polymer design using connectivity indices. *Industrial and engineering chemistry research* **1999**, *38*, 1884–1892.

[298] Raman, V. S.; Maranas, C. D. Optimization in product design with properties correlated with topological indices. *Computers and chemical engineering* **1998**, *22*, 747–763.

[299] Siddhaye, S.; Camarda, K. V.; Topp, E.; Southard, M. Design of novel pharmaceutical products via combinatorial optimization. *Computers and Chemical Engineering* **2000**, *24*, 701–704.