Accepted Manuscript

Title: Integration of modular process simulators under the Generalized Disjunctive Programming framework for the structural flowsheet optimization

Author: Miguel A. Navarro-Amorós Rubén Ruiz-Femenia
José A. Caballero

PII: S0098-1354(14)00095-7
DOI: <http://dx.doi.org/doi:10.1016/j.compchemeng.2014.03.014>
Reference: CACE 4927

To appear in: *Computers and Chemical Engineering*

Received date: 31-7-2013
Revised date: 23-3-2014
Accepted date: 25-3-2014

Please cite this article as: Navarro-Amorós, M. A., Ruiz-Femenia, R., & Caballero, J. A., Integration of modular process simulators under the Generalized Disjunctive Programming framework for the structural flowsheet optimization, *Computers and Chemical Engineering* (2014), <http://dx.doi.org/10.1016/j.compchemeng.2014.03.014>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Integration of modular process simulators under the Generalized Disjunctive Programming framework for the structural flowsheet optimization

Miguel A. Navarro-Amorós, Rubén Ruiz-Femenia, José A. Caballero*

Department of Chemical Engineering. University of Alicante. Ap. Correos 99. 03080 Alicante. Spain

* Corresponding author. Tel.: +34 965 903400x2322; fax: +34 965 903826. E-mail address: caballer@ua.es (J.A. Caballero).

Highlights

A new modeling framework that exploits the synergistic combination of commercial process simulators and GDP models

Our methodology allows to include easily logical relationships among alternatives

The proposed tool uses a logic based Outer Approximation algorithm

The methodology is applied to the synthesis of a methanol plant where different alternatives

Abstract

The optimization of chemical processes where the flowsheet topology is not kept fixed is a challenging discrete-continuous optimization problem. Usually, this task has been performed through equation based models. This approach presents several problems, as tedious and complicated component properties estimation or the handling of huge problems (with thousands of equations and variables). We propose a GDP approach as an alternative to the MINLP models coupled with a flowsheet program. The novelty of this approach relies on using a commercial modular process simulator where the superstructure is drawn directly on the graphical use interface of the simulator. This methodology takes advantage of modular process simulators (specially tailored numerical methods, reliability, and robustness) and the flexibility of the GDP formulation for the modeling and solution. The optimization tool proposed is successfully applied

to the synthesis of a methanol plant where different alternatives are available for the streams, equipment and process conditions.

Keywords

Process synthesis, Generalized Disjunctive Programming, Modular simulators, logic-based optimization algorithm

1. Introduction

One common approach for the optimization of real chemical process handles continuous process parameters (temperatures, pressures, flowrates, compositions, etc.) as the unique optimization variables while the flowsheet topology is kept fixed. A popular tool to perform this task are the process simulators based on the modular architecture, which are perfectly suited for simulation problems but loses part of its attractiveness for optimization or synthesis problems. In addition, chemical process synthesis also demands to make decisions related to process topology, which implies the inclusion of integer variables as free variables in the model, leading to a Mixed-Integer Nonlinear Programming (MINLP) problem (Lorenz T. Biegler et al., 1997; Ignacio E. Grossmann, 2002). This fact presents both opportunities and challenges for researchers to develop new tools that facilitate the synthesis of chemical plants to chemical engineers.

The Generalized Disjunctive Programming (GDP) modeling framework introduced by Raman and Grossmann (1994) has brought to Process System Engineering (PSE) community the powerful framework of the disjunctive programming, which was originally developed by Balas (1979, 1998) as an alternative representation of mixed-integer programming problems. GDP allows to model chemical plant synthesis problems through the use of higher level of logic constructs (Hooker & Osorio, 1999; Raman & Grossmann, 1994) that make the formulation step more intuitive and systematic, retaining in the model the underlying logical structure of the problem. GDP represents problems in terms of Boolean and continuous variables, allowing the

representation of constraints as algebraic equations, disjunctions and logic propositions (Beaumont, 1990). The development of GDP in the chemical engineering community has led to the development of customized algorithms that exploit this alternative modeling framework. In particular, Turkay and Grossmann (1996) extended the outer approximation (OA) algorithm (Duran & Grossmann, 1986) for MINLPs into a logical-equivalent algorithm. Later, Lee and Grossmann (2000) developed a disjunctive branch and bound.

GDP techniques have been successfully incorporated to many types of PSE optimization problems such as process flowsheet synthesis, design of distillation columns, scheduling and design of batch processes. In 1996, Turkay and Grossmann published a paper in which they proposed a GDP algorithm for structural flowsheet optimization problem and tested on several examples, including the synthesis of a vinyl chloride monomer process consisting of 32 units. Process synthesis with heat integration was also solved using disjunctions and logic propositions by Grossmann and coworkers (1998). One year later, Caballero and Grossmann (1999) reported an aggregated model for the synthesis of heat-integrated distillation columns modeled as a generalized disjunctive program. Later, a disjunctive programming model was also applied to the synthesis of distillation column sequences (Yeomans & Grossmann, 2000). In all these works, the problem is entirely described on explicit equations by a general modeling language system, like GAMS (Rosenthal 2013), and usually relies on simplified models (i.e., shortcut or aggregated methods) for the unit operations in the flowsheet and for the prediction of the physical properties of the components (e.g., for the vapor-liquid equilibrium). The first feature of this approach leads to difficulties, when modeling the problem, in the initialization step, which may converted into a daunting task. On the other hand, the use of simplified models for the unit operations could be not accurate enough to capture key aspects of a real chemical process plant. Moreover, using simplified physical property models can predict inaccurate thermodynamic properties, leading to misleading results.

The disadvantages listed in the last paragraph can be overcome by incorporating process simulators to the synthesis problem. Flowsheeting software provides realistic simulations and hence an optimal solution closer to the real implementation as they offer tailored numerical techniques developed for converging the different units and provides an extensive component database and reliable physical property methods. The usage of chemical process simulators as an implicit model for solving synthesis problems through a MINLP approach is not new. Harsh et al. (1989) developed an interface with a MINLP and FLOWTRAN, for the retrofit of an ammonia

process. Diwekar et al (Diwekar et al., 1992) proposed a MINLP synthesizer using Aspen Plus. Diaz and Bandoni (1996) used a MINLP formulation with an existing ad-hoc process simulator for the optimization of a real ethylene plant. Caballero et al. (2005) proposed a superstructure-based optimization algorithm for the rigorous design of distillation columns that combines a process simulator (Aspen HYSYS) with explicit equations. Latter Brunet et al. (2012) used the same algorithm for the optimization of an ammonia-water absorption cooling cycle implemented in Aspen Plus. Flowsheet process optimization with heat integration has also been performed using an hybrid simulation optimization approach, in which the process is solved by a commercial process simulator (Aspen HYSYS), and the heat integration model is in equation form (Navarro-Amorós et al., 2013). All these works are based on the augmented penalty/equality relaxation outer-approximation algorithm (Viswanathan & Grossmann, 1990). Other process simulators (SuperPro) has also been coupled with a multi-objective Matlab optimizer (Taras & Woinaroschy, 2012).

Another approach for the synthesis problem combines process simulators with metaheuristic algorithms. Although metaheuristic algorithms are not able to guarantee the optimality of the solutions found, they can find solutions for some real-world problems that exhibit high levels of complexity (Gendreau et al., 2010). Perhaps the most serious disadvantages of metaheuristic algorithms are that the number of function evaluations to converge could be large, and as well as they exhibit poor performance in highly constrained systems. A considerable amount of literature supports the integration of a process simulator with an external optimizer based on metaheuristic algorithms. Gross and Roosen (1998) demonstrated the suitability of a genetic algorithm coupled with the process simulator Aspen Plus to optimize arbitrary flowsheets. Leboeiro and Acevedo (2004) also succeeded in problems where deterministic mathematical algorithms had failed, using an optimization framework for the synthesis of complex distillation sequences based on a modified GA coupled with Aspen Plus. The same combination of process simulator and metaheuristic algorithm is adopted by Vazquez-Castillo et al. (2009) to address the optimization of five distillation sequences. Subsequent works used a multiobjective GA (Gutiérrez-Antonio & Briones-Ramírez, 2009) for the optimization of thermally coupled distillation systems (Bravo-Bravo et al., 2010; Cortez-Gonzalez et al., 2012; Gutiérrez-Antonio et al., 2011), and for the retrofit of a subcritical pulverized coal power plant with an MEA-based carbon capture and CO₂ compression system (Eslick & Miller, 2011). Finally, Odjo et al. (2011) also presented a general

framework for the synthesis of chemical processes using a hybrid approach with Hysys and genetic algorithms.

In this paper we present a new modeling framework for dealing with superstructure-based synthesis problems that exploits the synergistic combination of commercial process simulators with GDP formulation and their corresponding logic-based solution algorithms. As far as these authors know, it has not been reported a simulation-optimization tool for solving the synthesis of chemical plants whose superstructure is drawn directly on the process simulator graphical user interface (GUI). We achieve this aim by developing a GDP modeling system that interfaces with a process simulator (Aspen Hysys) at the NLP step to optimize the structure and parameters of a methanol plant based on a superstructure which involves alternative equipment, process conditions and stream configurations. Our methodology allows easily including soft constraints and logical relationships among alternatives, which ensure feasible solutions. The proposed tool uses the logic based Outer Approximation algorithm and hence it is not required to reformulate the problem as an MINLP.

The remainder of this article is organized as follows. The problem statement is first formally expressed. Then the methodology is introduced. In this section, the logic based outer approximation algorithm, integration of the process simulator in the algorithm and the connection with the external optimization solver are described. The proposed simulation-optimization framework is illustrated through a case study based on a methanol plant in the next section, where the superstructure and the disjunctions are presented. In this section, the results are also briefly described. Finally, we draw the conclusions from this work.

2. Problem statement

Given a superstructure for the synthesis of chemical process plant, with some specifications fixed, determine the optimal process flowsheet that leads to the maximum value of an economic indicator. The solution must include both topological and operational (temperatures, pressures, flow rates) information.

3. Methodology

We have developed a modeling system with the following characteristics:

1. The complete modeling system is developed in Matlab (MATLAB., 2006.).
2. Indexing capacities for both algebraic equations and implicit models.
3. Use of Boolean variables, disjunctions and logic propositions. Allowing the direct formulation of the problem as a disjunctive problem without MINLP reformulation.
4. Interfaced with different commercial solvers for NLP, LP, MILP models through Matlab-Tomlab (Holmström et al., 2010), and with homemade implementations of the logic based Outer Approximation algorithm (Turkay & Grossmann, 1996).
5. Communication with process simulators and other third party models, except those developed in Matlab, is done by the Windows COM capabilities.

Figure 1 shows a scheme of the tool we have implemented within Matlab environment with the Aspen HYSYS process simulator embedded, and allowing the user to model the optimization problem under the principles of GDP. To illustrate how our tool is used, we have added a video in the supplementary material section that describes the entire process step by step.

Figure 1.

3.1. Generalized Disjunctive Programming vs discrete-continuous simulation-optimization approach

The main purpose of the GDP simulation-optimization framework developed is to avoid some of the problems that arise in the “classical” simulation-optimization approach, where the chemical process synthesis problem is posed as a discrete-continuous optimization problem, and then formulated as an MINLP. The algorithms for MINLPs start by solving a relaxed problem (usually an integer relaxation), in which integer (binary) variables are assumed to be continuous. This relaxed problem presents some difficulties:

1. It is common that zero flows appear in some streams. The behavior of unit operations is simulator dependent and even for the same process simulator different units show dissimilar responses. In some cases everything works nicely, the zero flows do not affect the unit, but in other cases an error is dispatched and therefore the complete optimization fails. Setting the lower bound to a low value (e.g., 1×10^{-5}) is not always working, some

units require a minimum flow and again an error is thrown if the minimum flow is not reached. Even though if this last approach works, it must be taken into account in the model formulation that usually force the variables associated to a given unit operation to be zero if that unit does not exist.

2. All the units must be present in the initial NLP optimization (and in all others), even in the case of sub-problems in which a sub-set of units do not exist, which slow down the optimization.

Another disadvantage associated with the MINLP approach, as pointed out by Reneaume et al (1995), is that there are implicit relations among the interest variables calculated by the simulator and both continuous decision (independent variables) and binary topological variables. Reneaume et al. (1995) proposed using new "pseudo variables" and "pseudo-torn streams". Alternatively, it is possible to perform a mapping between internal variables calculated by the process simulator and a set of new external variables (that can be considered also as 'independent variables') breaking in that way those implicit relationships. The relation between external and internal variables are forced only if the unit in which those variables appear exists, but this also means to 'relax' that mapping in the initial relaxed problem which has two major consequences. First, the total number of independent variables increases, and second the relaxation gap is also worsened.

The "classical" simulation-optimization approach also entails two additional drawbacks. First, as the size of the MINLP problem increases, the increase in the size of the master and subproblems could become excessive for a reasonable computational performance. And second, singularities due to linearizations at zero flows and non convexities can cut off the global optimum (Türkyay & Grossmann, 1996).

Hence, a better solution strategy for flowsheet optimization problems would be advisable to address the difficulties arising in MINLP formulations. Accordingly, we use a GDP formulation with a logic based solver that leads to the following advantages:

1. The Outer Approximation Logic Based algorithm (and all its modifications) does not solve a relaxed problem but just a set of sub-problems that correspond to feasible flowsheets.
2. From a formal point of view the variables of a non-existing unit in a NLP sub-problem are forced to be zero, but in the practical implementation all those units are discarded (they

do not appear in the flowsheet) so the problem related to zero flows of non-existing units is completely avoided.

3. The NLPs are smaller because only existing units (those whose corresponding Boolean variable are true) are included in the flowsheet, at difference with the MINLP approach, whereas inactive units do not appear in the flowsheet.
4. The size of the MILP master problems are also smaller because only the linearizations of the existing units at each major iteration are included.
5. The implicit relations between continuous decision and independent variables do not appear (in other words, we are not adding linearizations in non-existing units as the MINLP approach does).

3.2. Logic Based Outer Approximation algorithm with an embedded process simulator

As mentioned above, we use the Logic-based Outer Approximation algorithm (Türkay & Grossmann, 1996) to fully exploit the structure of the GDP representation of our problem. The Logic-Based OA shares the main idea of the traditional OA for MINLP, which is to solve iteratively a master problem given by a linear GDP, leading to a lower bound of the solution (z^{LB}), and an NLP subproblem, which provides an upper bound (z^{UB}). The general structure of a nonconvex GDP formulation is as follows:

$$\begin{aligned}
 \min_{\mathbf{x}, \mathbf{Y}_{ik}} \quad & z = f(\mathbf{x}) \\
 \text{s.t.} \quad & \mathbf{h}(\mathbf{x}) = 0 \\
 & \mathbf{g}(\mathbf{x}) \leq 0 \\
 & \bigvee_{i \in D_k} \begin{bmatrix} \mathbf{Y}_{ik} \\ \mathbf{r}_{ik}(\mathbf{x}) \leq 0 \\ \mathbf{s}_{ik}(\mathbf{x}) = 0 \end{bmatrix} \quad \mathbf{k} \in K \\
 & \Omega(\mathbf{Y}) = \text{True} \\
 & \mathbf{x}^{lo} \leq \mathbf{x} \leq \mathbf{x}^{up} \\
 & \mathbf{x} \in \mathbb{R}^n, \mathbf{Y}_{ik} \in \{\text{True}, \text{False}\}, i \in D_k, k \in K
 \end{aligned} \tag{1}$$

where $\mathbf{x}$ is a vector of continuous variables representing pressures, temperatures and flow rates of the streams in a process flowsheet superstructure. The objective function is normally a cost function of the continuous variables. The common equality set of constraints $\mathbf{h}(\mathbf{x}) = 0$ represents equipment rating equations and mass and energy balances; and the common set of

inequalities $g(x) \leq 0$, represent design specifications. Both sets of constraints must hold true regardless the discrete decisions. The underlying alternatives in the superstructure are represented in the continuous space by a set of disjunctions $k \in K$, each of which contains $i \in D_k$ terms. Each term of the disjunction represents the potential existence of an equipment (or stream) i for performing a processing task k , and has associated a Boolean variable Y_{ik} and a set of constraints $r_{ik}(x) \leq 0$ and $s_{ik}(x) = 0$, which are normally associated with the investment and operations cost, the energy and mass balance, and physical and chemical equilibrium for the particular equipment i . When the term is not active ($Y_{ik} = \text{False}$), the corresponding constraints are ignored. Finally, the symbolic equation $\Omega(Y) = \text{True}$ represents the set of logic propositions that relates the Boolean variables, which in PSE generally indicates the logic implications among the equipment to define a feasible topology for the process flowsheet.

It is noteworthy that for the general GDP formulation (1), the terms in the disjunctions are linked by an inclusive OR logical operator (i.e., $A \vee B$ is true if A or B or both are true). Nevertheless, we have implemented a logic-based OA algorithm that requires to reformulate the disjunctive part of the problem as a set of special type of disjunctions, each of which contains only two terms and in one of them all the variables are set to zero. Fortunately, any disjunction in its general form has a straightforward reformulation to the special 2 terms disjunction (see Appendix 1).

As we are dealing with a process simulator embedded in a GDP formulation, it is convenient to define a partition of x into dependent x^D and independent (or design) variables x^I . The latter is the set of optimization variables and its dimension is equal to the degrees of freedom of the nonlinear problem obtained when the binary variables are fixed. By this partition the common equality constraint $h(x)$ can be solved for the dependent variables x^D given a vector of independent variables x^I , $x^D = \tilde{h}(x^I)$. In an analogous manner, for each equipment i assigned to a task k the dependent variables associated to it can be expressed as functions of the decision variables $x^D = \tilde{s}_{ik}(x^I)$. In this work, dependent variables x^D cannot explicitly written in terms of decision variables, but they are implicitly calculated at the process simulator, and then are used at the optimization level to evaluate the objective function and the common and particular constraints. Accordingly, the GDP problem (1) can be rewritten as:

$$\begin{aligned}
 & \min_{x^D, x^I, Y_{ik}} \quad z = f(x^D, x^I) \\
 & \text{s.t.} \quad x^D = h^{\text{Implicit}}(x^I) \\
 & \quad h(x^D, x^I) = 0 \\
 & \quad g(x^D, x^I) \leq 0 \\
 & \quad \left[\begin{array}{l} Y_{ik} \\ x^D = s_{ik}^{\text{Implicit}}(x^I) \\ \bigvee_{i \in D_k} r_{ik}(x^D, x^I) \leq 0 \\ s_{ik}(x^D, x^I) = 0 \end{array} \right] \quad k \in K \\
 & \quad \Omega(Y) = \text{True} \\
 & \quad x^{I,lo} \leq x^I \leq x^{I,up} \\
 & \quad x^D \in \mathbb{R}^{n_D}, x^I \in \mathbb{R}^{n_I}, Y_{ik} \in \{\text{True}, \text{False}\}, i \in D_k, k \in K
 \end{aligned} \tag{2}$$

Note that in (2) as we introduce dependent variables in explicit equations (for example in $h(x^D, x^I) = 0$ or in $g(x^D, x^I) \leq 0$), a sequential function evaluation is required, first the implicit models are solved and then the explicit constraints can be evaluated.

3.2.1. Logic-based NLP subproblem

For fixed values of the Boolean variables Y_{ik} (i.e., given a flowsheet configuration) the corresponding NLP subproblem is as follows:

$$\begin{aligned}
 & \min_{x^D, x^I} \quad z^{\text{UB}}(Y_{ik}) = f(x) \\
 & \text{s.t.} \quad x^D = h^{\text{Implicit}}(x^I) \\
 & \quad h(x^D, x^I) = 0 \\
 & \quad g(x^D, x^I) \leq 0 \\
 & \quad \left. \begin{array}{l} x^D = s_{ik}^{\text{Implicit}}(x^I) \\ r_{ik}(x^D, x^I) \leq 0 \\ s_{ik}(x^D, x^I) = 0 \\ x^{I,lo} \leq x^I \leq x^{I,up} \end{array} \right\} \text{for } Y_{ik} = \text{True}, i \in D_k, k \in K \\
 & \quad x^D \in \mathbb{R}^{n_D}, x^I \in \mathbb{R}^{n_I}, Y_{ik} \in \{\text{True}, \text{False}\}, i \in D_k, k \in K
 \end{aligned} \tag{3}$$

It is worthy to emphasize that only the constraints that belong to the selected equipment or stream (i.e., associate Boolean variable $Y_{ik}^1 = \text{True}$) are imposed. This leads to a substantial reduction in the size of the NLP subproblem compared to the direct application of the traditional OA method on the MINLP reformulation.

3.2.2. Logic-based Master problem

Assuming that L NLP subproblems are solved in which sets of linearizations are generated for the objective function and the common constraints and particular constraints in the subsets of disjunction terms $L_{ik} := \{l : Y_{ik}^l = \text{True}\}$, we define the following disjunctive OA master problem:

$$\begin{aligned}
 \min_{x^I, Y_{ik}} \quad & z^{LB} = \alpha \\
 \text{s.t.} \quad & \alpha \geq f(x^{D,l}, x^{I,l}) + \nabla_{x^I} f(x^{D,l}, x^{I,l})^T (x^I - x^{I,l}) \\
 & t^{h,l} \left[h(x^{D,l}, x^{I,l}) + \nabla_{x^I} h(x^{D,l}, x^{I,l})^T (x^I - x^{I,l}) \right] \leq 0 \\
 & g(x^{D,l}, x^{I,l}) + \nabla_{x^I} g(x^{D,l}, x^{I,l})^T (x^I - x^{I,l}) \leq 0 \quad \left. \vphantom{\begin{aligned} \alpha \geq f(x^{D,l}, x^{I,l}) + \nabla_{x^I} f(x^{D,l}, x^{I,l})^T (x^I - x^{I,l}) \\ t^{h,l} \left[h(x^{D,l}, x^{I,l}) + \nabla_{x^I} h(x^{D,l}, x^{I,l})^T (x^I - x^{I,l}) \right] \leq 0 \\ g(x^{D,l}, x^{I,l}) + \nabla_{x^I} g(x^{D,l}, x^{I,l})^T (x^I - x^{I,l}) \leq 0 \end{aligned}} \right\} \quad l = 1, \dots, L \\
 & \bigvee_{i \in D_k} \left[\begin{array}{c} Y_{ik} \\ r_{ik}(x^{D,l}, x^{I,l}) + \nabla_{x^I} r_{ik}(x^{D,l}, x^{I,l})^T (x^I - x^{I,l}) \leq 0 \\ t^{s,l} \left[s_{ik}(x^{D,l}, x^{I,l}) + \nabla_{x^I} s_{ik}(x^{D,l}, x^{I,l})^T (x^I - x^{I,l}) \right] \leq 0 \end{array} \right] \quad l \in L_{ik} \quad k \in K \\
 & \Omega(Y) = \text{True} \\
 & x^{I,lo} \leq x \leq x^{I,up} \\
 & x^D \in \mathbb{R}^{n_D}, \alpha \in \mathbb{R}^1, Y_{ik} \in \{\text{True}, \text{False}\}, i \in D_k, k \in K
 \end{aligned} \tag{4}$$

where t^h and t^s are assigned with either the value 1, 0 or -1 , depending the sign of the Lagrange multiplier of the corresponding nonlinear constraint. The constraints of a particular disjunction term (equipment i for task k) are only included in the master problem if the corresponding Boolean variable Y_{ik} is True, whereas linearizations of temporally inactive terms are simply discarded. Again this property constitutes a major difference to the standard OA method. Note that the master MILP (4) problem is not a function of the dependent variables.

3.2.3. MILP reformulation of the master problem

The master problem of the logic-based OA algorithm can be reformulated as an MILP using either Big-M (BM) or Hull Reformulation (HR) formulations. We apply the tighter formulation, that is HR, and then the disjunctions of the GDP master problem are reformulated as follows:

$$\left. \begin{aligned} & \left. \begin{aligned} & r_{ik}(x^{D,l}, x^{I,l})y_{ik} + \nabla_{x^I} r_{ik}(x^{D,l}, x^{I,l})^T (x^{I,i} - x^{I,l}) \leq u_{ik}^r \\ & t_{ik}^{s,l} \left[s_{ik}(x^{D,l}, x^{I,l})y_{ik} + \nabla_{x^I} s_{ik}(x^{D,l}, x^{I,l})^T (x^{I,i} - x^{I,l}) \right] \leq u_{ik}^s \end{aligned} \right\} l \in L_{ik}, i \in D_k \\ & x^I = \sum_{i \in D_k} x^{I,i} \\ & x^{I,lo} y_{ik} \leq x^{I,i} \leq x^{I,up} y_{ik}, i \in D_k \end{aligned} \right\} k = 1, \dots, K \quad (5)$$

where the superscripts lo and up denote the lower and upper bounds, respectively. Each vector of independent variables x^I related to a potential equipment i is disaggregated into as many new vector variables $x^{I,i}$ as the number of alternative equipments (or streams) for potentially performing task k . The upper and lower bounds applied for all the disaggregated variables with the binary variables y_{ik} are used to force the variables to zero when the mode i is not selected for performing task stage k .

The exclusive OR logic operator in the disjunction part of the GDP master problem (4) is transformed into the following linear constraint:

$$\sum_{i \in D_k} y_{ik} = 1, \quad k = 1, \dots, K \quad (6)$$

Furthermore, we add a set of binary cuts (Balas & Jeroslow, 1972) to exclude the previous solution for the binary variables:

$$\sum_{(i,k) \in B^l} y_{ik} - \sum_{(i,k) \in N^l} y_{ik} \leq |B^l| - 1, \quad l = 1, \dots, L \quad (7)$$

where B^l is the subset defined for each NLP subproblem that stores the binary variables y_{sm} with a value of 1, and N^l is the subset that collects the remaining binary variables for that NLP subproblem; i.e., $B^l = \{(s,m) : Y_{sm}^l = \text{True}\}$ and $N^l = \{(s,m) : Y_{sm}^l = \text{False}\}$.

To avoid infeasible master problems caused by the nonconvexity of the GDP problem (1), we relax the linearized constraints in (5) by introducing positive slack variables u_{ik}^r and u_{ik}^s , respectively. As the common constraints in Eq. (2) can be also nonlinear, we add the slack variables u^h and u^g to the RHS of the common linearized constraints in Eq. (4). These slack

variables are included in the objective function through a penalty term with weights w_{ik}^r , w_{ik}^s , w^h and w^g chosen to be sufficiently large. Accordingly, the objective function of the Master Problem is rewritten as:

$$\min_{x^l, Y_{ik}} Z_{LB}^L = \alpha + \sum_i \sum_k (w_{ik}^r)^T u_{ik}^r + \sum_i \sum_k (w_{ik}^s)^T u_{ik}^s + (w^h)^T u^h + (w^g)^T u^g \quad (8)$$

Some important remarks deserve special attention. GDP algorithms require convexity to guaranty convergence to a global optimal solution. In an implicit model it is difficult to prove convexity, even in the case the model be convex, but in general we must assume non convexity. Therefore, there is no guarantee to find a global solution, and only locally optimal solutions should be expected.

3.3. Connection between Matlab and Aspen Hysys

The developers of Aspen HYSYS also followed the paradigm shift in the development of process simulators, from procedural to objected-oriented programming. Accordingly, Aspen HYSYS is programmed with 32-bit C++ (Bhutani, 2007), which gives it the ability to lend its functionalities to be used in other application software. That makes Aspen HYSYS a very powerful and useful tool in the design of our hybrid framework. We use the binary-interface standard Component Object Model (COM), by Microsoft, to interact with Aspen HYSYS through the objects exposed by the developers of Aspen HYSYS. We utilize Matlab as an automation client to access these objects and interact with Aspen HYSYS, which works as an automation server (see Figure 1). By writing Matlab code, it is possible to send and receive information to and from the process simulator. Thus, the exposed objects make possible to perform nearly any action that is accomplished through the Aspen HYSYS graphical user interface, allowing us to use Aspen HYSYS as a calculation engine. According to the objected-oriented programming nomenclature (Booch et al., 2007), in Aspen Hysys, the functions defined within each object (e.g., reactor) are called methods (i.e., governing equations), and the variables contained in an object are known as properties (e.g., feed composition).

3.4. Connection between Matlab and Optimizer

We use the TOMLAB optimization environment, which provides an interface between the Matlab model and the available optimization solvers. This tool allows us to standardize the model definition and then use all the available solvers regardless the different syntax required for each solver. We do not need to make a specific interface routine for each optimization solver. We use the CPLEX version 12.2.0.0 solver for the MILP problems and the CONOPT solver for the NLP subproblems. The latter solver is based on the Generalized Reduced Gradient (GRD), which is suitable for models where feasibility is difficult to achieve.

As mentioned above, the modeling framework proposed does not require to rewrite the problem as an MINLP, allowing for direct application of solution methods to problems formulated as GDP. To this aim, we implement the logic based OA algorithm with the special feature that allows to use also implicit models (i.e., models inside a process simulators). An implicit model can be treated as a black box with a rigid input-output structure whose derivative information is not available. The models in a modular chemical process simulator are accurate enough for simulation purposes, but they could introduce some numerical noise (i.e., the solution varies slightly with identical initial values) that prevent the accurate determination of derivative information (L. T. Biegler & Hughes, 1982). We capture the gradient information by a finite difference approach with a perturbation size that balances and minimizes the error due to noise and the error in the approximation of the Jacobian. As the perturbation size increases, the error in the approximation of the Jacobian becomes significant. On the other hand, the response of a small perturbation may be corrupted by convergence noise. Here, it is appropriate to mention that the numerical noise effect is magnified by recycles in the flowsheet, because they behave as “error accumulators” (Martín, 2014). In this case, instead of using the simulator utilities to converge the recycles we connect the simulator with the external NLP solver to converge them. In that way, we have a complete control over the numerical methods used for convergence of recycles. Furthermore, although, both the number of variables handles by the NLP solver and the number of explicit equality constraints increases, in general the model is more robust and usually the computational time does not increase because it is not required to converge all the recycles each time the simulator is called.

4. CASE STUDY

As an example to illustrate the correct behavior of the proposed methodology, we present the case of the synthesis of methanol. This process has been studied extensively in the past (D.A. Bell et al., 2010; Ghiotti & Boccuzzi, 1987; Klier, 1982; Kung, 1980; Lange, 2001; Luyben, 2010; Skrzypek et al., 1994). Figure 2 shows a simplified flowsheet of the methanol process using syngas as feed stream. Conventional methanol production uses a feed stock of reformed methane that contains hydrogen, carbon monoxide and carbon dioxide in a ratio of $N_{H_2} / (2N_{CO} + 3N_{CO_2})$ close to the stoichiometric ratio of unity. The chemistry of the methanol process involves a lot of reactions, but only three reactions are significant, the synthesis of methanol from carbon monoxide (R-1), the synthesis of methanol from carbon dioxide (R-2), and the water gas shift reaction (R-3).

Figure 2. .

The objective of this example is to maximize the profit of the process. We consider two available feeds with different characteristics and prize (See Table 2). We also consider two products in the process, a principal (and desired) product with a high sale prize (0.25 €/kg) and a subproduct (purge stream) with a low sale price (0.018 €/kg). The equipment cost is calculated using correlations from the literature, and to this end we use the correlations given by Turton et al. (2008), and also the prize of all used utilities are obtained from this reference. Finally, we update the prices to 2012 using the "Chemical Engineering Plant Cost Index" (CEPCI). The annual

cost of the equipment is calculated for a time horizon (n) of 10 years and an interest rate per year (i) of 8% (Smith, 2005) using the following expression,

$$\text{Annualized capital cost} = \text{capital cost} \cdot \frac{i(1+i)^n}{(1+i)^n - 1} \quad (9)$$

The simulation is performed using the sequential modular simulator Aspen-HYSYS with the Soave-Redlich-Kwong (SRK) equation of state and default values.

To check the capabilities of our simulation-optimization tool, we build a superstructure of the methanol process that includes all the alternatives of interest (Figure 3). Note that the aim of this example is to demonstrate the behavior of the methodology when different alternatives exist for each of unit operations (disjunctions). The key of this process is the reactor operation. The formation of methanol, as a typical heterogeneously catalyzed reaction, can be described by absorption-desorption mechanism (Langmuir-Hinshelwood or Eley-Rideal). The synthesis of methanol is a pressure and temperature dependent process. The carbon monoxide and carbon dioxide conversions up to attainment of equilibrium are shown as a function of pressure and temperature in Table 1.

Table 1.

In order to clearly illustrate the problem, for the sake of simplicity, but without loss of generality we will focus on two operating conditions, one specifically working at 200°C and 50 atm (called “Low Conversion Condition”) and the other working at 200°C and 100 atm (Called “High Conversion Condition”). Note that this problem can be formulated for the reactor operating in any combination of pressure and temperature, which increases the number of alternatives.

Figure 3.

The superstructure presents in Figure 3 include the following alternatives:

Feed stream

Two different synthesis gas streams are available, both containing the reactants H_2 , CO_2 and CO , and a small amount of inert, CH_4 . The characteristics of different feed streams are shown in Table 2. For this example, we consider the option to select only one feed stream. This is modeled with the following OR exclusive disjunction:

$$\left[\begin{array}{c} Y_1 \\ \text{Feed Cost} = \tau \gamma_A^F \overline{M}_A^F F^{\text{Feed}} \\ F_A^{\text{Feed, LB}} \leq F^{\text{Feed}} \leq F_A^{\text{Feed, UB}} \end{array} \right] \vee \left[\begin{array}{c} Y_2 \\ \text{Feed Cost} = \tau \gamma_B^F \overline{M}_B^F F^{\text{Feed}} \\ F_B^{\text{Feed, LB}} \leq F^{\text{Feed}} \leq F_B^{\text{Feed, UB}} \end{array} \right] \quad (10)$$

where F^{Feed} is the molar flow rate of the synthesis gas feed stream selected. $F_A^{\text{Feed, UB}}$ and $F_B^{\text{Feed, UB}}$, or $F_A^{\text{Feed, LB}}$ and $F_B^{\text{Feed, LB}}$ are the upper and lower bounds on the availability of each feed stream; $\overline{M}_A^F$ and $\overline{M}_B^F$ are the average molecular weight of the syngas feed streams of type A and B, respectively; γ_A^F and γ_B^F are the costs of the syngas of type A and B, respectively; and we consider 8000 h of operation per year (τ).

Table 2.

As commented above, in the logic based outer approximation, we need two term disjunctions in which one of them forces all the variables to be zero (or simply to be discarded in the NLP problem). This can be done as follows. In the Appendix 1 we include a general reformulation for an n-term disjunction. For this particular case we reformulate (10) as follows:

$$\begin{aligned}
 & \left[\begin{array}{c} Y_1 \\ \text{Feed Cost}_A = \tau_A \gamma_A^F \overline{M}_A^F F_A^{\text{Feed}} \\ F_A^{\text{Feed, LB}} \leq F_A^{\text{Feed}} \leq F_A^{\text{Feed, UB}} \end{array} \right] \vee \left[\begin{array}{c} \neg Y_1 \\ \text{Feed Cost}_A = 0 \\ F_A^{\text{Feed}} = 0 \\ \tau_B = 0 \end{array} \right] \\
 & \left[\begin{array}{c} Y_2 \\ \text{Feed Cost}_B = \tau_B \gamma_B^F \overline{M}_B^F F_B^{\text{Feed}} \\ F_B^{\text{Feed, LB}} \leq F_B^{\text{Feed}} \leq F_B^{\text{Feed, UB}} \end{array} \right] \vee \left[\begin{array}{c} \neg Y_2 \\ \text{Feed Cost}_B = 0 \\ F_B^{\text{Feed}} = 0 \\ \tau_B = 0 \end{array} \right] \quad (10) \\
 & \tau = \tau_A + \tau_B \\
 & F^{\text{Feed}} = F_A^{\text{Feed}} + F_B^{\text{Feed}} \\
 & \text{Feed Cost} = \text{Feed Cost}_A + \text{Feed Cost}_B \\
 & Y_1 \vee Y_2
 \end{aligned}$$

The rest of disjunctions are also reformulated as two term disjunctions even though it is not explicitly stated in the text.

Feed Compression system

The feed enters the process at low pressure (20 atm) and must be compressed to a higher pressure where reaction is feasible (in this case, we consider two possible operation conditions, 50 or 100 atm). For compression, we assume the choice between a single compressor (single stage compression) or a system consisting of two compressor with intermediate cooling (two-stage compression). Furthermore, in each compression system, the gas pressure can be increased to 50 bar or to a higher value of 100 bar. To model the latter choice for the single-stage compression alternative, we define the Boolean variables Y_3 and Y_4 to operate either at low or high output pressure respectively, and write following disjunction:

$$\left[P^{\text{FC, out}} = 50 \text{ atm} \right] \vee \left[P^{\text{FC, out}} = 100 \text{ atm} \right] \quad (11)$$

where $P^{\text{FC, out}}$ is the pressure of the stream leaving the feed compression system (In Figure 3 corresponds with the pressure of the stream leaving compressor K-100). For the case of two-stage compression with intermediate cooling, we also formulate a disjunction with two terms, one corresponding to the low pressure (Y_5) and the other for the high pressure (Y_6):

$$\left[\begin{array}{c} Y_5 \\ P^{FC,out} = 50 \text{ atm} \\ 30 \text{ atm} \leq P^{FC,intermediate} \leq 50 \text{ atm} \\ 40 \text{ }^\circ\text{C} \leq T^{FC,intermediate} \leq 70 \text{ }^\circ\text{C} \end{array} \right] \vee \left[\begin{array}{c} Y_6 \\ P^{FC,out} = 100 \text{ atm} \\ 40 \text{ atm} \leq P^{FC,intermediate} \leq 64 \text{ atm} \\ 40 \text{ }^\circ\text{C} \leq T^{FC,intermediate} \leq 60 \text{ }^\circ\text{C} \end{array} \right] \quad (12)$$

where $P^{FC,intermediate}$ and $T^{FC,intermediate}$ are the intermediate pressure and temperatures, respectively (In Figure 3, $P^{FC,intermediate}$ corresponds with the pressure of the stream leaving compressor K-101, and $T^{FC,intermediate}$ corresponds with the temperature of the stream leaving heat exchanger E-100) and $P^{FC,out}$ is the pressure of the stream leaving the feed compression system (In Figure 3 corresponds with the pressure of the stream leaving compressor K-102).

In the single stage compression (11) and the two-stage compression system (12) disjunctions, we link the two terms of each disjunction with the so-called logical operator “at most one” (a variation of the OR operator equivalent to $\neg A \vee \neg B$) to not force to select one of the two terms (i.e. the two Boolean variables associated with the low and high pressure can be simultaneously false).

To avoid the combination with the four Boolean variables (Y_1 , Y_2 , Y_3 and Y_4) being simultaneously false, which has no physical meaning as the gas feed must be compressed, we add the following constraint to our optimization problem:

$$Y_3 \vee Y_4 \vee Y_5 \vee Y_6 \quad (13)$$

Reactor + flash units

The gas reaction (Eqs. R-1, R-2 y R-3) takes place in a high conversion expensive reactor or in a less expensive reactor working at lower conversion. The difference between them concerns the pressure at which the reactions are produced. While the expensive reactor works at 100 atm (high conversion), the other works at 50 atm (low conversion). To model the latter choice reactor alternative, we define the Boolean variables Y_7 and Y_8 to operate either at high or low conversion conditions respectively. The characterization of each reactor is totally defined by the specification of degree of conversion of the different compounds (CO and CO₂). Note that the reaction R-3 is negligible under these operating conditions as compared with the other reactions. The reactor is

cooled using water at ambient conditions. The capital cost of the reactor depends on its volume (for a detailed description of the volume calculation see appendix 2).

The next step in the process is the separation system. The vapor stream leaving the reactor contains the desired product (methanol) and high concentration of light components, as CO or CO₂. Therefore, a flash tank is used to remove most of the light components and obtain methanol with desired composition. The combination of pressure and temperature required in the flash unit for the desired methanol purity (molar fraction > 90%) is attained by an expansion valve and a water-cooled heat exchanger. Note that the lower and upper bounds of the pressure in flash unit are assigned to the minimum (pressure of feed stream) and the maximum pressure of the system (pressure in the reactor), respectively. Furthermore, the lower bound of temperature in flash unit is 40°C because of the use of water as refrigerant in heat exchanger, and the upper bound is 140°C due to higher values do not allow to reach the desired product composition.

The choice of reactor and flash conditions are formulated with the following OR exclusive disjunction:

$$\left[\begin{array}{c} Y_7 \\ X_{1,CO} = 0.99 \\ X_{2,CO_2} = 0.83 \\ 20 \leq P^{FLASH} \leq 100 \\ 40 \leq T^{FLASH} \leq 140 \end{array} \right] \vee \left[\begin{array}{c} Y_8 \\ X_{1,CO} = 0.96 \\ X_{2,CO_2} = 0.29 \\ 20 \leq P^{FLASH} \leq 50 \\ 40 \leq T^{FLASH} \leq 140 \end{array} \right] \quad (14)$$

where $X_{1,CO}$ and X_{2,CO_2} are the degree of conversion of CO and CO₂ in reactions R-1 and R-2, respectively. P^{FLASH} corresponds with the pressure of the stream leaving expansion valve V-100 for disjunction Y_7 and valve V-101 for disjunction Y_8 , and T^{FLASH} corresponds with the pressure of the stream leaving heat exchanger E-101 for disjunction Y_7 and valve E-103 for disjunction Y_8 .

Heating/cooling before Reactor

The two operating conditions in the reactor, previously selected, implies that the temperature of its inlet stream must be 200°C. To get this, the resulting stream from the sum of compressed feed stream and the recycled stream must be heated or cooled. In this case, and after a previously sensitivity study, we know that only in the case of using the single compressor at 100

atm, the temperature of the stream exceeds 200°C, and must be cooled. In all other cases, the stream is lower than 200°C, and must be heated.

To model this situation and guide the system to the correct choice, we define the Boolean variables Y_{13} and Y_{14} to select a heater or a cooler, respectively. All the previous situations can be formulated as Booleans expressions:

Single stage compression until 50 atm implies heating: $Y_3 \Rightarrow Y_{13}$

Single stage compression until 100 atm implies cooling: $Y_4 \Rightarrow Y_{14}$

Two-stage compression with intermediate cooling until 50 atm implies heating: $Y_5 \Rightarrow Y_{15}$

Two-stage compression with intermediate cooling until 100 atm implies heating: $Y_6 \Rightarrow Y_{16}$

Note that the specification of the temperature of the outlet stream of the heat exchanger (200°C) is specified in the simulators. To simulate the cooler, we use a water-cooled heat exchanger using water at ambient conditions as refrigerant. To simulate the heater, we use a heat exchanger using high pressure steam as hot utility.

Recycled stream compression system

The recycled stream in the process must be compressed until the operation pressure of the selected reactor (50 or 100 atm). As in the feed compression system, we assume the choice between a single compressor (single stage compression) or a system with two compressor with intermediate cooling (two-stage compression). Furthermore, in each system, the gas pressure can be increased to 50 bar or to a higher value of 100 bar. To model the latter choice for the single-stage compression alternative, we define the Boolean variables Y_9 and Y_{10} to operate either at low or high output pressure respectively and we use the following disjunction:

$$\left[\begin{array}{c} Y_9 \\ P^{RC,out} = 50 \text{ atm} \end{array} \right] \vee \left[\begin{array}{c} Y_{10} \\ P^{RC,out} = 100 \text{ atm} \end{array} \right] \quad (15)$$

where $P^{RC,out}$ is the pressure of the stream leaving the feed compression system (In Figure 3 corresponds with the pressure of the stream leaving compressor K-103).

For the case of two-stage compression with intermediate cooling, we also formulate a disjunction with two terms, one corresponding to the low pressure (Y_{11}) and the other for the high pressure (Y_{12}):

$$\left[\begin{array}{c} Y_{11} \\ P^{RC,out} = 50 \text{ atm} \\ 30 \text{ atm} \leq P^{RC,intermediate} \leq 50 \text{ atm} \\ 40 \text{ }^\circ\text{C} \leq T^{RC,intermediate} \leq 70 \text{ }^\circ\text{C} \end{array} \right] \vee \left[\begin{array}{c} Y_{12} \\ P^{RC,out} = 100 \text{ atm} \\ 40 \text{ atm} \leq P^{RC,intermediate} \leq 64 \text{ atm} \\ 40 \text{ }^\circ\text{C} \leq T^{RC,intermediate} \leq 60 \text{ }^\circ\text{C} \end{array} \right] \quad (16)$$

where $P^{RC,intermediate}$ and $T^{RC,intermediate}$ are the intermediate pressure and temperatures, respectively (In Figure 3, $P^{RC,intermediate}$ corresponds with the pressure of the stream leaving compressor K-105, and $T^{RC,intermediate}$ corresponds with the temperature of the stream leaving heat exchanger E-104) and $P^{RC,out}$ is the pressure of the stream leaving the feed compression system (In Figure 3 corresponds with the pressure of the stream leaving compressor K-104).

As in feed compression system, the two disjunctions require the following logic propositions between the Boolean variables to ensure that at most one term is selected in each disjunction:

$$\begin{aligned} \text{at most } (Y_9, Y_{10}) \text{ is equivalent to } [\neg Y_9 \vee \neg Y_{10}] \\ \text{at most } (Y_{11}, Y_{12}) \text{ is equivalent to } [\neg Y_{11} \vee \neg Y_{12}] \end{aligned} \quad (17)$$

To avoid the combination with the four Boolean variables (Y_9 , Y_{10} , Y_{11} and Y_{12}) being simultaneously false, which has no physical meaning as the gas feed must be compressed, we add the following constraint to our optimization problem:

$$Y_9 \vee Y_{10} \vee Y_{11} \vee Y_{12} \quad (18)$$

Apart from the disjunctions, the superstructure has other important characteristics which must be commented:

Vent-to-recycle Split

An important characteristic of this example is the vent-to-recycle split. The vapor stream leaving the flash unit contains both useful compounds (reactants as H_2 , CO and CO_2) and waste products (inerts, CH_4 and H_2O). In this situation, the split ratio has an important effect in the global process. While higher flows in recycled stream increase the recovery and reuse of reactants, the

concentrations of inert products in the system also increase, and as results, the compression costs increase. In addition, high flows of vent stream reduce the compression cost in the system, but increase the losses of reactants (hydrogen, carbon monoxide, and carbon dioxide).

In our case study, we define the split variable as an independent variable, which is controlled by the optimizer.

Results

The optimal configuration is shown in Figure 4 and the computational results are shown in Table 3.

Figure 4.

Table 3.

Table 4.

As we discussed above, the first step in the methodology is the initialization of all the units inside the disjunctions. In this case, this consists in selecting a minimum set of feasible flowsheets in such a way that all the terms in the disjunctions be true at least once. In this example, and the compression system entails 2 disjunctions (with 4 terms or alternatives in total), we need to solve 4 initial subproblem to cover all the terms. Then, a Master problem is solved to provide a new set of Boolean variables that produce better results than in previous solution. From Master results, we solved a NLP problem to obtain the better feasible solution. To avoid that the algorithm stops early due to the non-convex constraints, we use a stopping criterion based on the heuristic: stop as two consecutive NLP subproblem worsen. For our particular case, the optimal solution is found at the initial problem 2 with an objective value of 57,55 MM €/year.

The optimal solution selects the low conversion reactor. In this case, the low cost in the compression system (50 atm) compensate the lower conversion obtained with this reactor. The main characteristics of the selected equipment are shown in Table 4.

5. CONCLUSIONS

We show that a process synthesis problem can be addressed under the perspective of the General Disjunctive Programming (GDP) framework (without an MINLP reformulation), and solved by the logic based outer approximation algorithm with a commercial process simulator embedded. Conceptually the GDP approach facilitates the model formulation for the final user retaining in the model the underlying logical structure of the problem. We propose a novel approach that combines the flexibility of the GDP formulation with the benefits of the commercial process simulators (i.e., rigorous models for the estimation of the thermophysical properties). The novelty of the proposed framework relies on the advantage that the superstructure of the process is directly built in the graphical user interface of the simulator, and on the fact that the GDP approach avoids some of the drawbacks of the “classical” simulation-optimization method, in which the process synthesis problem is posed as a discrete-continuous problem and then reformulated as an MINLP. The proposed approach is illustrated through a case study for the production of methanol, where some constraints are fixed and we establish several alternatives for some streams, tasks (one and two stages compression system; two types of reactors), and process conditions (low and high pressures). We also confirm that GDP simulation-optimization approach provides an intuitive way to synthesize chemical processes. Finally, we illustrate our tool with a video that shows for the methanol case study the complete automatic implementation of the GDP model directly over the process simulator in which units are dynamically connected and disconnected (see supplementary material).

Acknowledgements

The authors wish to acknowledge support from the Spanish Ministry of Science and Innovation (CTQ2012-37039-C02-02).

1. REFERENCES

- Balas, E. (1979). Disjunctive Programming. *Annals of Discrete Mathematics*, 5, 3-51.
- Balas, E. (1998). Disjunctive programming: Properties of the convex hull of feasible points. *Discrete Applied Mathematics*, 89, 3-44.
- Balas, E., & Jeroslow, R. (1972). Canonical Cuts on the Unit Hypercube. *SIAM Journal on Applied Mathematics*, 23, 61-69.
- Beaumont, N. (1990). An algorithm for disjunctive programs. *European Journal of Operational Research*, 48, 362-371.
- Bell, D. A., Towler, B. F., & Fan, M. (2010). *Coal Gasification and Its Applications*: Elsevier Science.
- Bell, D. A., Towler, B. F., Fan, M., & Books24x7 Inc. (2011). *Coal gasification and its applications*. In (1st ed.). Oxford, U.K. ; Burlington, Mass.: William Andrew.
- Bhutani, N. (2007). Modeling, simulation and multi-objective optimization of industrial hydrocrackers. National University of Singapore.
- Biegler, L. T., Grossmann, I. E., & Westerberg, A. W. (1997). *Systematic methods of chemical process design*. Upper Saddle River, N.J.: Prentice Hall PTR.
- Biegler, L. T., & Hughes, R. R. (1982). Infeasible path optimization with sequential modular simulators. *AIChE Journal*, 28, 994-1002.
- Booch, G., Maksimchuk, R. A., Engle, M. W., Young, B. J., Conallen, J., & Houston, K. A. (2007). *Object-oriented analysis and design with applications* (3rd ed.). Upper Saddle River, NJ: Addison-Wesley.
- Bravo-Bravo, C., Segovia-Hernández, J. G., Gutiérrez-Antonio, C., Durán, A. L., Bonilla-Petriciolet, A., & Briones-Ramírez, A. (2010). Extractive dividing wall column: Design and optimization. *Industrial and Engineering Chemistry Research*, 49, 3672-3688.
- Brunet, R., Cortés, D., Guillén-Gosálbez, G., Jiménez, L., & Boer, D. (2012). Minimization of the LCA impact of thermodynamic cycles using a combined simulation-optimization approach. *Applied Thermal Engineering*, 48, 367-377.
- Caballero, J. A., & Grossmann, I. E. (1999). Aggregated models for integrated distillation systems. *Industrial and Engineering Chemistry Research*, 38, 2330-2344.
- Caballero, J. A., Milan-Yanez, D., & Grossmann, I. E. (2005). Rigorous Design of Distillation Columns: Integration of Disjunctive Programming and Process Simulators. *Industrial & Engineering Chemistry Research*, 44, 6760-6775.
- Cortez-Gonzalez, J., Segovia-Hernández, J. G., Hernández, S., Gutiérrez-Antonio, C., Briones-Ramírez, A., & Rong, B. G. (2012). Optimal design of distillation systems with less than N-1 columns for a class of four component mixtures. *Chemical Engineering Research and Design*, 90, 1425-1447.
- Díaz, M. S., & Bandoni, J. A. (1996). A mixed integer optimization strategy for a large scale chemical plant in operation. *Computers and Chemical Engineering*, 20, 531-545.
- Diwekar, U. M., Grossmann, I. E., & Rubin, E. S. (1992). An MINLP process synthesizer for a sequential modular simulator. *Industrial & Engineering Chemistry Research*, 31, 313-322.
- Duran, M., & Grossmann, I. (1986). An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36, 307-339.
- Eslick, J. C., & Miller, D. C. (2011). A multi-objective analysis for the retrofit of a pulverized coal power plant with a CO₂ capture and compression process. *Computers & Chemical Engineering*, 35, 1488-1500.
- Gendreau, M., Potvin, J.-Y., & SpringerLink (Online service). (2010). *Handbook of metaheuristics*. In *International series in operations research & management science* v. 146 (2nd ed., pp. 1 online resource (xix, 648 p.)). New York: Springer.
- Ghiotti, G., & Boccuzzi, F. (1987). Chemical and Physical Properties of Copper-Based Catalysts for CO Shift Reaction and Methanol Synthesis. *Catalysis Reviews*, 29, 151-182.
- Gross, B., & Roosen, P. (1998). Total process optimization in chemical engineering with evolutionary algorithms. *Computers & Chemical Engineering*, 22, Supplement 1, S229-S236.
- Grossmann, I. E. (2002). Review of Nonlinear Mixed-Integer and Disjunctive Programming Techniques. *Optimization and Engineering*, 3, 227-252.
- Grossmann, I. E., Yeomans, H., & Kravanja, Z. (1998). A rigorous disjunctive optimization model for simultaneous flowsheet optimization and heat integration. *Computers and Chemical Engineering*, 22, S157-S164.

- Gutiérrez-Antonio, C., Briones-Ramírez, A., & Jiménez-Gutiérrez, A. (2011). Optimization of Petlyuk sequences using a multi objective genetic algorithm with constraints. *Computers and Chemical Engineering*, 35, 236-244.
- Gutiérrez-Antonio, C., & Briones-Ramírez, A. (2009). Pareto front of ideal Petlyuk sequences using a multiobjective genetic algorithm with constraints. *Computers and Chemical Engineering*, 33, 454-464.
- Harsh, M. G., Saderne, P., & Biegler, L. T. (1989). A mixed integer flowsheet optimization strategy for process retrofits-the debottlenecking problem. *Computers and Chemical Engineering*, 13, 947-957.
- Holmström, K., Göran, A. O., & Edvall, M. M. (2010). USER'S GUIDE FOR TOMLAB 7. In (pp. 268).
- Hooker, J. N., & Osorio, M. A. (1999). Mixed logical-linear programming. *Discrete Applied Mathematics*, 96-97, 395-442.
- Klier, K. (1982). Methanol synthesis. *Adv. Catal*, 31, 243-313.
- Kung, H. H. (1980). Methanol Synthesis. *Catalysis Reviews*, 22, 235-259.
- Lange, J.-P. (2001). Methanol synthesis: a short review of technology improvements. *Catalysis Today*, 64, 3-8.
- Leboreiro, J., & Acevedo, J. (2004). Processes synthesis and design of distillation sequences using modular simulators: a genetic algorithm framework. *Computers & Chemical Engineering*, 28, 1223-1236.
- Lee, S., & Grossmann, I. E. (2000). New algorithms for nonlinear generalized disjunctive programming. *Computers and Chemical Engineering*, 24, 2125-2141.
- Luyben, W. L. (2010). Design and Control of a Methanol Reactor/Column Process. *Industrial & Engineering Chemistry Research*, 49, 6150-6163.
- Martín, M. (2014). Introduction to Software for Chemical Engineers: CRC Press, In Press (ISBN 9781466599369).
- MATLAB. (2006.). The Language of Technical Computing. In: The Mathworks Inc.
- Mäyrä, O., & Leiviskä, K. (2008). Modelling in methanol synthesis In (Vol. Report A No 37, pp. 43). Control engineering laboratory: University of OULU.
- Navarro-Amorós, M. A., Caballero, J. A., Ruiz-Femenia, R., & Grossmann, I. E. (2013). An alternative disjunctive optimization model for heat integration with variable temperatures. *Computers and Chemical Engineering*, 56, 12-26.
- Odjo, A. O., Sammons, N. E., Yuan, W., Marcilla, A., Eden, M. R., & Caballero, J. A. (2011). Disjunctive-genetic programming approach to synthesis of process networks. *Industrial and Engineering Chemistry Research*, 50, 6213-6228.
- Raman, R., & Grossmann, I. E. (1994). Modelling and computational techniques for logic based integer programming. *Computers and Chemical Engineering*, 18, 563-578.
- Reneaume, J. M. F., Koehret, B. M., & Joulia, X. L. (1995). Optimal process synthesis in a modular simulator environment: New formulation of the mixed-integer nonlinear programming problem. *Industrial and Engineering Chemistry Research*, 34, 4378-4394.
- Rosenthal, R. E. (2013). GAMS: A user's guide. GAMS Development Corporation, Washington, DC, USA. In.
- Skrzypek, J., Słoczyński, J., Słoczyński, J., & Ledakowicz, S. (1994). Methanol synthesis: science and engineering: Polish Scientific Publishers.
- Smith, R. M. (2005). Chemical Process: Design and Integration: John Wiley & Sons.
- Taras, S., & Woinaroschy, A. (2012). An interactive multi-objective optimization framework for sustainable design of bioprocesses. *Computers and Chemical Engineering*, 43, 10-22.
- Turkay, M., & Grossmann, I. E. (1996). Logic-based MINLP algorithms for the optimal synthesis of process networks. *Computers & Chemical Engineering*, 20, 959-978.
- Turkay, M., & Grossmann, I. E. (1998). Structural flowsheet optimization with complex investment cost functions. *Computers and Chemical Engineering*, 22, 673-686.
- Türkay, M., & Grossmann, I. E. (1996). Logic-based MINLP algorithms for the optimal synthesis of process networks. *Computers and Chemical Engineering*, 20, 959-978.
- Turton, R., Bailie, R. C., Whiting, W. B., & Shaeiwitz, J. A. (2008). Analysis, Synthesis and Design of Chemical Processes: Pearson Education.
- Vazquez-Castillo, J. A., Venegas-Sánchez, J. A., Segovia-Hernández, J. G., Hernández-Escoto, H., Hernández, S., Gutiérrez-Antonio, C., & Briones-Ramírez, A. (2009). Design and optimization, using genetic algorithms, of intensified distillation systems for a class of quaternary mixtures. *Computers & Chemical Engineering*, 33, 1841-1850.
- Viswanathan, J., & Grossmann, I. E. (1990). A combined penalty function and outer-approximation method for MINLP optimization. *Computers & Chemical Engineering*, 14, 769-782.
- Yeomans, H., & Grossmann, I. E. (2000). Disjunctive programming models for the optimal design of distillation columns and separation sequences. *Industrial and Engineering Chemistry Research*, 39, 1637-1648.

APPENDIX 1. Reformulation of a N term disjunction in N two term disjunctions

Consider the N term disjunction ($N = \text{card}(D)$ disjunction)

$$\bigvee_{i \in D} \left[\begin{array}{c} Y_i \\ h_i(x) = 0 \\ g_i(x) \leq 0 \\ x_i^{Lo} \leq x \leq x_i^{Up} \end{array} \right]$$

it can be rewritten as N disjunctions with two terms each as follows:

$$\left[\begin{array}{c} Y_i \\ h_i(z_i) = 0 \\ g_i(z_i) \leq 0 \\ x_i^{Lo} \leq z_i \leq x_i^{Up} \end{array} \right] \bigvee \left[\begin{array}{c} \neg Y_i \\ z_i = 0 \end{array} \right] \quad i \in D$$

$$x = \sum_{i \in D} z_i$$

APPENDIX 2. Calculation of reactor volume

The equipment cost is calculated using correlations from the literature, and to this end we use the correlations given by Turton et al. (Turton et al., 2008). The capital cost of the reactor is calculated based on his volume. This parameter is calculated as follow. For a continuous stirred flow reactor (CSTR) a molar flow balance for component j gives the volume of the reactor V . Thus, for the hydrogen component, the resulting expression to calculate the volume of the reactor is:

$$V = \frac{F_{H_2} - F_{H_2}^{in}}{r_{H_2}} \quad (19)$$

where r_{H_2} is the rate of disappearance of H_2 and F_{H_2} is the molar flow rate of H_2 at the output stream, which is calculated according to the following expression:

$$F_j = F_j^{\text{in}} + \sum_i \alpha_{ij} \frac{F_k^{\text{in}} X_{ik}}{(-\alpha_{ik})}, \quad \forall j, k \in KC_i \quad (20)$$

where α_{ij} are the stoichiometric coefficient for component j in reaction i , and X_{ik} assesses the extent of reaction i with respect a key component k for that reaction (KC_i is the subset of components that contains the key component k for the reaction i). Specifically, the conversions of the reaction R-1 with respect carbon monoxide and reaction R-2 with respect to carbon dioxide are:

$$\begin{aligned} X_{1,\text{CO}} &= \frac{F_{\text{CO}}^{\text{in}} - F_{\text{CO}}}{F_{\text{CO}}^{\text{in}}} \\ X_{2,\text{CO}_2} &= \frac{F_{\text{CO}_2}^{\text{in}} - F_{\text{CO}_2}}{F_{\text{CO}_2}^{\text{in}}} \end{aligned} \quad (21)$$

The reaction rate with respect H_2 term in Eq. (19) is calculated from reaction rates with respect to methanol for both reactions ($r_{\text{CH}_3\text{OH},i}$) according to the following expression:

$$r_{\text{H}_2} = a^{\text{cat}} \sum_i \alpha_{i,\text{H}_2} \frac{r_{\text{CH}_3\text{OH},i}}{(-\alpha_{\text{CH}_3\text{OH},i})} \quad (22)$$

where a^{cat} is the mass of the catalyst per unit reactor volume, $1.154 \text{ ton cat} \cdot \text{m}^{-3}$ (Mäyrä & Leiviskä, 2008).

The rate for the methanol synthesis from carbon monoxide reaction with respect to methanol ($r_{\text{CH}_3\text{OH},1}$) is obtained with the following kinetic model (D. A. Bell et al., 2011), which uses partial pressures of the components (P_j):

$$r_{\text{CH}_3\text{OH},1} = \frac{k_1 K_{\text{CO}} \left[P_{\text{CO}} P_{\text{H}_2}^{3/2} - \frac{P_{\text{CH}_3\text{OH}}}{K_1^{\text{eq}} P_{\text{H}_2}^{1/2}} \right]}{\left(1 + K_{\text{CO}} P_{\text{CO}} + K_{\text{CO}_2} P_{\text{CO}_2} \right) \left(P_{\text{H}_2}^{1/2} + k_2 P_{\text{H}_2\text{O}} \right)} \quad (23)$$

where $P_j = (F_j/F)P$ for all j , and K_1^{eq} is the equilibrium constants for the synthesis of methanol from carbon monoxide R-1 reaction. In Eq. Error! Reference source not found. the kinetic constants k_1 , k_2 , K_{CO} and K_{CO_2} are calculated as follows:

$$k_1 = 2.69 \times 10^7 \exp\left(\frac{-109900}{RT}\right) \quad (24)$$

$$k_2 = 4.13 \times 10^{-11} \exp\left(\frac{-104500}{RT}\right) \quad (25)$$

$$K_{CO} = 7.99 \times 10^{-7} \exp\left(\frac{-58100}{RT}\right) \quad (26)$$

$$K_{CO_2} = 1.02 \times 10^{-7} \exp\left(\frac{67400}{RT}\right) \quad (27)$$

where R is the gas law constant in $J \cdot mol^{-1} \cdot K^{-1}$ and T temperature in K.

There is an analogous expression for the rate of reaction R-2 with respect to methanol:

$$r_{CH_3OH,2} = \frac{k_3 K_{CO_2} \left[P_{CO} P_{H_2}^{3/2} - \frac{P_{CH_3OH} P_{H_2O}}{K_2^{eq} P_{H_2}^{3/2}} \right]}{\left(1 + K_{CO} P_{CO} + K_{CO_2} P_{CO_2} \right) \left(P_{H_2}^{1/2} + k_2 P_{H_2O} \right)} \quad (28)$$

where K_2^{eq} is the equilibrium constant for the synthesis of methanol from carbon dioxide R-2 reaction, and k_3 is the kinetic constant evaluated with the expression:

$$k_3 = 4.36 \times 10^2 \exp\left(\frac{-65200}{RT}\right) \quad (29)$$

Knowing the volume of the reactor, we calculate the capital cost of the reactor is calculated using the correlations provides by Turton et al.

Figure Captions

Figure 1. Scheme of the optimization modeling framework.

Figure 2. Methanol simplified process flowsheet.

Figure 3. Superstructure for the Methanol synthesis process in Aspen-Hysys.

Figure 4. Optimal configuration of the case study.

Table 1. Temperature and pressure dependence of the carbon monoxide and carbon dioxide equilibrium conversions

Temp (°C)	CO conversion*			CO ₂ conversion		
	50 atm	100 atm	300 atm	50 atm	100 atm	300 atm
200	96,3	99,0	99,9	28,6	83,0	99,5
250	73,0	90,6	99,0	14,4	45,1	92,4
300	25,4	60,7	92,8	14,1	22,3	71,0
350	-2,3	16,7	71,9	9,8	23,1	50,0
400	- 12,8	-7,3	34,1	27,7	29,3	41,0

* Negative sign denote CO formation via Equation (R-3): $\text{CO} + \text{H}_2\text{O} \leftrightarrow \text{CO}_2 + \text{H}_2$

Table 2. Synthesis gas streams data.

Feed Stream	Min Molar Flow (kmol/s)	Max Molar Flow (kmol/s)	P (atm)	Cost (€/kg)	Composition			
					H ₂	CO	CO ₂	CH ₄
Feed A	0,05	1,44	20	0,026	0,70	0,26	0,02	0.02
Feed B	0,05	1,89	20	0,02	0,75	0,17	0,05	0.03

Table 3. Computational Results

N° of Boolean variables			14	
N° of independent variables			29	
N° of linear explicit equations			51	
N° of non-linear explicit equations			12	
N° of implicit blocks			14	
Iterations *		Objective	CPU time (s)	Solver
Initialization4	NLP			
subproblems				
Initial	NLP	-560.64	22.20	CONOPT
Subproblem 1				
Initial	NLP	-575.46	12.40	CONOPT
Subproblem 2				
Initial	NLP	-562.63	12.28	CONOPT
Subproblem 3				
Initial	NLP	-574.49	7.52	CONOPT
Subproblem 4				
MILP Master, major iteration 1		110.54	0.09	CPLEX
NLP Subproblem 5		-573.82	19.97	CONOPT

* Pentium Dual-Core E5300 2.60GHz

Table 4. Results: Characteristic of equipment in optimal configuration

Compression System		Recycled Stream Compression System	
Type	Single Compression	Type	Single Compression
Inlet Pressure (atm)	20,0	Inlet Pressure (atm)	36,7
Compression ratio	2,5	Compression ratio	1,4
Power (MW)	6,662	Power (MW)	0,07
COST		COST	
Capital Cost (€)	822338	Capital Cost (€)	51006
Annualized Capital Cost (€/year)	122553	Annualized Capital Cost (€/year)	7601
Electricity Cost (€/year)	2309701	Electricity Cost (€/year)	24198
Heating or Cooling		Reactor System	
Type	Heater	Type	Low Conversion Reactor
Area (m ²)	85,6	Volume (m ³)	66,5
Inlet - Outlet Temp (°C)	143,4 - 200	Cold Utility	Water
Hot Utility	Steam HP(41atm)	Water flow (kg/h)	1527336
Steam Flow (kg/h)	6687,2	Energy (MW)	34,040
Energy (MW)	3,198	COST	
COST		Capital Cost (€)	49037
Capital Cost (€)	95792	Annualized Capital Cost (€/year)	7308
Annualized Capital Cost (€/year)	14276	Catalyst Cost (€/year)	173506
Utility Cost (€/year)	856503	Utility Cost (€/year)	328283
Flash System			
Type	Cooled water Heat exchanger		
Area (m ²)	636,5		
Inlet - Outlet Temp (°C)	196,7 - 59,9		
Cold Utility	Water		
Steam Flow (kg/h)	799778		
Energy (MW)	19,710		
COST			
Capital Cost (€)	191253		
Annualized Capital Cost	28502		

(€/year)

Utility Cost (€/year)

190083

Accepted Manuscript

Table Captions

Table 1. Temperature and pressure dependence of the carbon monoxide and carbon dioxide equilibrium conversions.

Table 2. Synthesis gas streams data.

Table 3. Computational Results.

Table 4. Results: equipment characteristics in the optimal configuration.

.

Figure(s)