Hybrid model generation for superstructure optimization with Generalized Disjunctive Programming

H. A. Pedrozo[a]; S. B. Rodriguez Reartes[a,b]; D. E. Bernal[d]; A. R. Vecchietti[c]; M. S. Diaz[a,b,*]; I. E. Grossmann[d]

[a] Planta Piloto de Ingeniería Química (PLAPIQUI CONICET-UNS), Camino La Carrindanga km. 7,
Bahía Blanca, Argentina
[b] Departamento de Ingeniería Química, Universidad Nacional del Sur (UNS),
Bahía Blanca, Argentina
[c] INGAR – Instituto de Desarrollo y Diseño (CONICET-UTN), Avellaneda 3657, Santa Fe,
Argentina
[d] Department of Chemical Engineering, Carnegie Mellon University, 5000 Forbes Avenue,
Pittsburgh, PA 15213, USA
[*]sdiaz@plapiqui.edu.ar

Abstract

We propose a novel iterative procedure to generate hybrid models (HMs) within an optimization framework to solve design problems. HMs are based on first principle and surrogate models (SMs) and they may represent potential plant units embedded within a superstructure. We generate initial SMs with simple algebraic regression models and refine them by adding Gaussian Radial Basis Functions in three steps: initial SM refinement, domain exploration, and, after solving the optimal design problem, further domain exploitation, until the convergence criterion is fulfilled. The superstructure optimization problem is formulated with Generalized Disjunctive Programming and solved with the Logic-based Outer Approximation algorithm. We addressed methanol synthesis and propylene plant design problems. Compared to rigorous model-based optimal design, the proposed HMs gave the same configuration, objective function and decision variables with maximum relative differences of 1 and 7 %, respectively. A sensitivity analysis shows that the proposed strategy reduced CPU time by 33 %.

Keywords: hybrid models; superstructure optimization; Logic-based Outer Approximation algorithm; GDP; propylene production; State equipment network

1 INTRODUCTION

The accurate modeling of a process flowsheet, including the detailed formulations of each unit, typically yields a large-scale nonconvex mixed-integer nonlinear programming (MINLP) problem. As MINLP formulations are NP-hard (Sahinidis, 2019), model developers usually have to employ tailored solution strategies based on the system knowledge. On the other hand, Generalized Disjunctive Programming (GDP) allows a more intuitive way of formulating a mathematical problem by introducing specific process information through disjunctive constraints and logical propositions. Furthermore, GDP formulations can be solved using powerful decompositions algorithms as the Logic-based Outer Approximation algorithm (L-bOA) (Chen et al., 2020; Türkay and Grossmann, 1996). However, in several cases, the analytical form of the objective function or constraints are not available since they are commonly obtained from simulators or computer codes (Bajaj et al., 2018). When the mathematical problem includes known and unknown constraints, it is referred to as hybrid or grey-box. In order to address these problems, surrogate models can be used. Surrogate modeling is a technique based on building an inexpensive function from input-output data (Bhosekar and Ierapetritou, 2018; Kim and Boukouvala, 2020). This approach has been successfully implemented in optimal process design problems (Caballero and Grossmann, 2008; Kong et al., 2016; Pedrozo et al., 2020). Further, surrogate models can also be used to replace rigorous unit models in large-scale nonconvex problems. In this way, the number of equations and problem complexity can be reduced. The approximation of nonlinearities and nonconvexities is a significant challenge in surrogate modeling, so the common basis functions are nonlinear. There are two classes of basis functions. The first one includes simple algebraic regression models such as polynomial and bilinear functions, while the second class of basis functions includes interpolating functions such as Radial Basis Functions (RBFs) and Kriging. These basis functions have specific mathematical properties, which can be appropriately exploited to generate SMs. In particular, regression functions are widely used for algebraic optimization since their low complexity allows their simple implementation within large-scale optimization problems, and RBFs show the capacity of accurate representations of highly nonlinear models (Fang et al., 2005). These SMs can replace complex mathematical models of equipment units, such as distillation columns and process reactors, reducing the number of equations and nonlinear terms. In this context, regression functions are suitable choices due to their features, but they may fail to represent the unit performance accurately. On the other hand, RBFs may introduce additional nonlinear terms. Therefore, the combination of both basis functions can be a promising option.

Cozad et al. (Cozad et al., 2014) developed automated learning of algebraic models for optimization (ALAMO), which is a software package to build surrogate models based on regression functions such as linear, polynomial, or rational, as well as radial basis functions (RBF), that are globally optimized. Wilson et al.(Wilson and Sahinidis, 2017) evaluated the constrained regression feature in ALAMO, and showed the efficiency of error maximization sampling as an exploration method. Another strategy to generate surrogate models is based on Artificial Neural Networks (ANN), as proposed by Henao and Maravelias (Henao and Maravelias, 2011, 2010). This approach can be used to represent highly nonlinear functions and scales well with an increasing number of output variables. Boukouvala and Floudas (Boukouvala and Floudas, 2017) developed Algorithms for Global Optimization of coNstrAined grey-box compUTational problems (ARGONAUT), which is an optimization framework to solve problems including known and unknown constraints. A particular feature of ARGONAUT is the selection of basis functions for surrogates, which can be interpolating or non-interpolating.

In this work, we propose a novel procedure for generating surrogate models based on regression and interpolating functions to take advantage of the strengths of each model class within an optimization framework. These surrogate models are included together with first-principle models to represent process units in GDP problems for chemical plant optimal design. We propose an iterative optimization framework to solve the resulting GDP constrained with a hybrid model. The proposed procedure is applied to two case studies: a methanol synthesis problem, and a propylene production via olefin metathesis plant design problem.

2 METHODOLOGY

The objective in this work is to propose an iterative procedure to generate models for chemical plant optimal design. The main idea is to represent process units through first principles and surrogate models embedded within a superstructure representation.

We propose surrogate models based on simple algebraic regression and interpolating functions. On the one hand, low complexity models are obtained through algebraic regression basis functions, which can be efficiently used in optimization approaches (Wilson and Sahinidis, 2017). On the other hand, interpolating functions can accurately represent every function pattern, giving exact outputs in interpolation points (Chen et al., 2014). In particular, we employ Gaussian Radial Basis Functions (GRBFs) due to their performance within optimization frameworks (Amouzgar and Strömberg, 2017; McDonald et al., 2007). Since a good representation of the analytical model is required in the entire domain, interpolation functions are suitable to fit the residues of the regression model, without significantly changing model predictions away from interpolation points. GRBFs exhibit this feature, and GRBF coefficients are obtained by solving linear systems of equations.

The following function is used to build surrogate models:

$$h^s(v) \; = \sum_{i=1}^{N} c_i h_i^a(v) + \sum_{i'=1}^{M} w_{i'} e^{-\gamma \|v - vs_{i'}\|_2^2} \tag{1}$$

where v are the surrogate input variables; c_i and w_i are model parameters to fit; N is the number of algebraic functions; M is the number of interpolation points; γ is the form factor of GRBFs; vs_i are interpolation points; $h_i^a(v)$ are simple algebraic regression functions.

Both first principle and generated surrogate models are included as constraints for the optimal design problem, which is formulated as the following Generalized Disjunctive Programming (GDP) problem:

$$min\, f(x) \tag{2.1}$$

$$g(x) \leq 0 \tag{2.2}$$

$$g^s(x) \leq 0 \tag{2.3}$$

$$\vee_{k\in K_j} \begin{bmatrix} Y_{jk} \\ h_{jk}(x) \leq 0 \\ h_{jk}^s(x) \leq 0 \end{bmatrix} \quad j \in J \tag{2.4}$$

$$x \in X \subseteq \mathbb{R}^n \tag{2.5}$$

$$Y_{jk} \in \{True, False\}^m \tag{2.6}$$

where $f(x)$ in (2.1) is the objective function, $g(x)$ in (2.2) is the set of linear and nonlinear constraints from first principles, and $g^s(x)$ in (2.3) correspond to the equations of the surrogate models. Regarding the disjunctive equations (2.4), Y_{jk} is the Boolean variable associated with the k-th term of the j-th disjunction, $h_{jk}(x)$ are algebraic constraints and $h_{jk}^s(x)$ are surrogate models.

To solve the resulting GDP, we apply the Logic-based Outer Approximation algorithm (L-bOA). This strategy proceeds by decomposing the GDP into reduced NLP subproblems and Master MILP problems, in this way avoiding "zero-flow" units and constraints related to non-existing units or streams. L-bOA was implemented in LOGMIP 1.0 (Vecchietti and Grossmann, 2000, 1999). However, since it is no longer available in LOGMIP 2.0/GAMS, we have coded a custom implementation of the decomposition algorithm in order to use updated versions of the NLP and MILP solvers. The code starts by performing a set covering step (Türkay and Grossmann, 1996) in which selected NLP subproblems are solved for nonlinear disjunctions in which selected units are fixed for a specific flowsheet structure. The solution of these NLP subproblems provide points to obtain linearizations for all nonlinear disjunctions in the model. Thus, a linear GDP is generated, whose Master MILP is in turn re-written with a Big-M reformulation.

The proposed iterative procedure to generate and refine surrogate models within an optimization framework includes the following steps: initial surrogate model generation; exploration (adaptive sampling); solving the hybrid GDP problem; and exploitation of the NLP solutions.

2.1 Initial surrogate model generation

As a first step, sampling points are generated through a simple filling space strategy. However, as most of the sampling methods were developed for hypercube spaces, there are some infeasible points where the feasible region is different from a hypercube. Therefore, we apply a filtering strategy that eliminates non-feasible sampling points from the initial sampling set in the next step. There are no direct sampling methods to fill the feasible region defined by the problem constraints to the best of our knowledge. Therefore, we perform an adaptive sampling strategy in the following step of the algorithm (see Section 2.2).

Algebraic regression functions are generated with the machine learning software ALAMO (Cozad et al., 2014) *),* which solves a global optimization problem to ε tolerance. These functions are used to perform an initial fit with low complexity basis functions, including linear, quadratic, and cubic monomials, binomials and quadratic binomials, trinomials, and constant terms. This machine learning software determines model complexity through the number of terms in the fitness metric used as the objective function. In this work, we select the Bayesian Information Criterion (BIC), which is the most widely used criterion in the literature (Wilson and Sahinidis, 2017). BIC establishes a trade-off between model errors and model complexity. Once an algebraic regression surrogate model is generated, the relative residuals are analyzed to decide if a first refinement step is to be carried out. Relative errors of sampling data are calculated and sampling points whose errors are larger than a tolerance (ε_0) are included as interpolation points for Gaussian Radial Basis Functions (GRBFs). It should be noted that residuals are fit through GRBF, so the surrogate is exact in the interpolation points. An example is given in Section 3.3.4.

2.2 Adaptive sampling (exploration step)

For further refinement, adaptive sampling is carried out to find domain regions where the model requires additional exploration. In this step, optimization problems are formulated through the Error Maximization Sampling (EMS) strategy (Cozad et al., 2014; Wilson and Sahinidis, 2017), as follows,

$$max\left(\frac{h^r(v)-h^s(v)}{h^r(v)}\right)^2 \tag{3.1}$$

$$g^r(v) \leq 0 \tag{3.2}$$

$$v^L \leq v \leq v^U \tag{3.3}$$

Where $h^r(v)$ is the true model output variable, and $g^r(v)$ are the true model constraints. The EMS is a mathematical problem that allows identifying the domain regions where the difference between the surrogate and the original model is the largest. It should be noted that in problem

(3), the algebraic equations of the true model are assumed to be known. However, in a black box case for the true model, a derivative-free optimization method can be used (Rios and Sahinidis, 2013). In this work, problem (3) is solved to local optimality. The adaptive sampling technique is extended to multivariable surrogate models since several output variables are estimated from the same input variable vector. For instance, input variables for the surrogate model of distillations columns are component molar flowrates and specific enthalpy in the feed stream, and top pressure, from which surrogate functions for the corresponding capital cost, reboiler duty, top and bottom temperatures are generated (four surrogate functions). EMS is applied to each surrogate function.

The exploration step stops when the relative error is less than a tolerance, or when a maximum number of EMS problems is solved as proposed by Wilson and Sahinidis (Wilson and Sahinidis, 2017). In this work, we consider a relative tolerance of 0.03, which is equivalent to an EMS objective value of roughly 0.01, and the maximum number of mathematical problems solved at each iteration is 50.

2.3 Exploitation step

When solving the optimal design problem as a GDP problem constrained with a hybrid model, NLP subproblems may be infeasible due to the lack of accuracy of the surrogate models in key regions of variable input domains (Kim and Boukouvala, 2020). The true unit model may also be infeasible when it is evaluated at the optimal solution of an NLP subproblem. In both cases, feasible sampling points must be obtained to exploit that region, and consequently, improve the surrogate model performance. Therefore, an optimization problem is formulated to minimize the square Euclidean norm of the difference between the NLP subproblem solution (v^*) and the surrogate model input variables (v) that satisfy the true unit model constraints, as follows,

$$min\|v^* - v\|_2^2 \tag{4.1}$$

$$g^r(v) \leq 0 \tag{4.2}$$

$$v^L \leq v \leq v^U \tag{4.3}$$

It should be noted that v^* are parameters for Problem (4) since their values are solutions of the current NLP subproblems. Problem (4) solution provides a feasible sampling point that is included in the training set, and consequently, it allows the region exploitation in case of infeasible problems.

In the case that the true model is of black-box type, the feasibility problem could be solved using derivative-free optimization methods (Rios and Sahinidis, 2013). It must be noted that these methods are efficient for low-dimensional models, as is the case of individual equipment unit models.

It should be pointed out that we did not address any method to reduce the feasible region so as to maintain its original form. While strategies that 'contract' the feasible region could improve the numerical performance, the globally optimal solution could be missed during the domain reduction if not performed rigorously. Otherwise, the exploitation step has the potential to refine the regions where the optimal solution is likely to be, without reducing the feasible space.

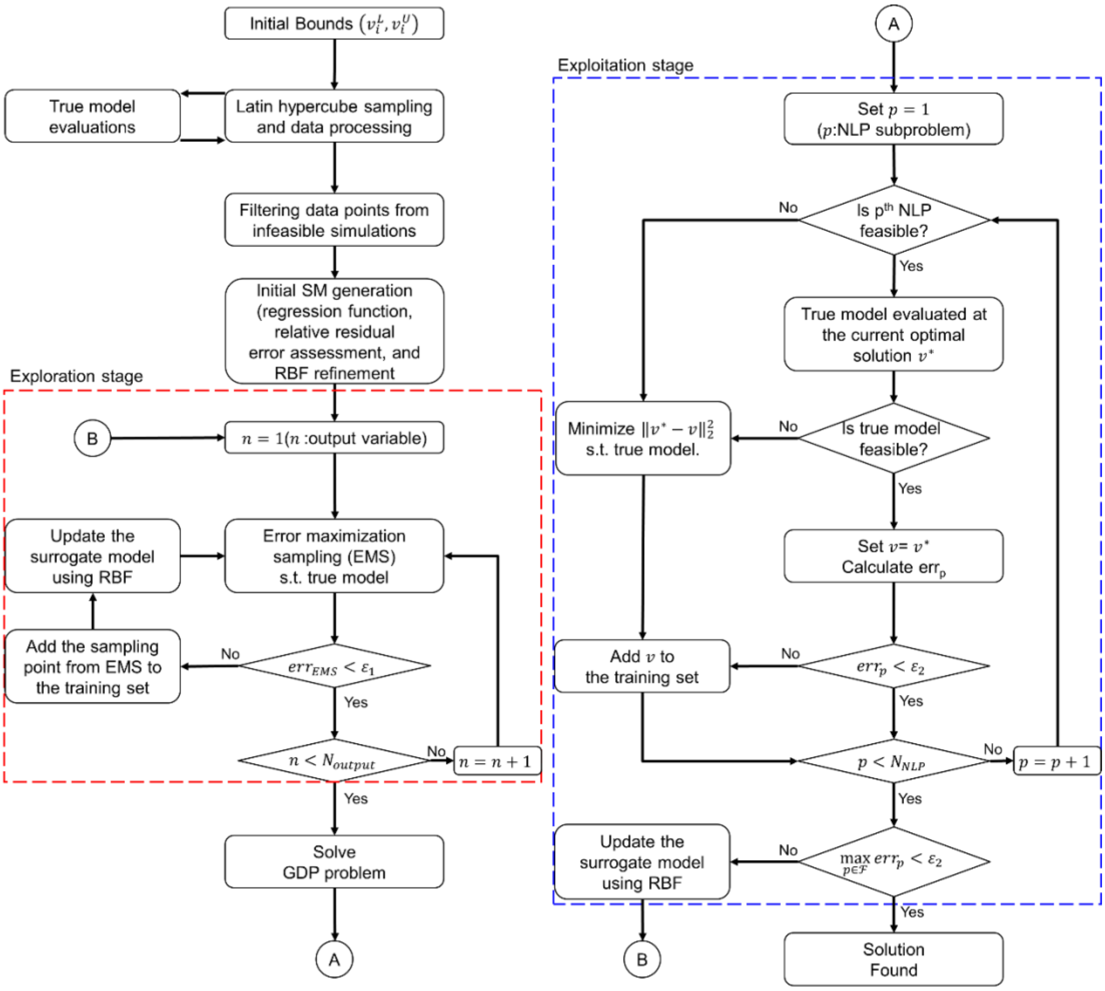
2.4 Solution strategy

In this work, we propose the iterative procedure for solving optimal design problems based on hybrid models shown in Fig. 1. A detailed description of this procedure is as follows:

1. Input variables bounds selection for each surrogate model.
2. Initial data generation with Latin Hypercube sampling (LHS) technique for function evaluations and determination of the corresponding output variables to build surrogate models.
3. Filtering step: To deal with constrained problems and the fact that some original unit model evaluations may be infeasible, a filtering strategy (Boukouvala and Floudas, 2017) is applied to disregard sampling data points that generate infeasible simulations.
4. Initial surrogate model (SM) generation
    4.1 SM generation with simple algebraic regression functions in the learning software ALAMO
    4.2 Relative residual error assessment to determine deviated sample points (DSP) in which these errors are larger than tolerance (ε_0)
    4.3 SM refinement with GRBFs for each DSP
5. Exploration step
    5.1 Error maximization sampling (EMS): Determination of the point where the SM has the least accuracy in the feasible region. The EMS problem is solved for each surrogate model output variable.
    5.2 SM update with GRBFs in the deviated sample points, which include additional points determined in 5.1
    5.3 Convergence criteria
6. Solution of GDP problem constrained by hybrid model, using the Logic-based Outer Approximation (L-bOA) algorithm.
7. Exploitation step

7.1 NLP subproblem solutions from each GDP provide feasible points for the domain region. Each process unit true model is simulated at these points and, in case feasible and low accuracy, they are included in the training set. Otherwise, a feasibility problem (4) is solved to obtain the nearest point to the solution to be included in the training set. In this way, the algorithm takes advantage of the L-bOA algorithm by testing and, eventually including, each NLP subproblem solution.

7.2 Convergence criteria: relative error of each output variable, of each surrogate model, evaluated at each feasible NLP subproblem solution must be less than a given tolerance (ε_2). If not satisfied, go to 5 (The number of major iteration of the algorithm is equal to the times the GDP problem is solved).



Initial Bounds (v_i^L, v_i^U)
True model evaluations
Latin hypercube sampling and data processing
Filtering data points from infeasible simulations
Initial SM generation (regression function, relative residual error assessment, and RBF refinement
Exploration stage
B
$n = 1$(n :output variable)
Update the surrogate model using RBF
Error maximization sampling (EMS) s.t. true model
Add the sampling point from EMS to the training set
No
$err_{EMS} < \varepsilon_1$
Yes
$n < N_{output}$
No
$n = n + 1$
Yes
Solve GDP problem
A
A
Exploitation stage
Set $p = 1$ (p:NLP subproblem)
No
Is pth NLP feasible?
Yes
True model evaluated at the current optimal solution v^*
Minimize $\|v^* - v\|_2^2$ s.t. true model.
No
Is true model feasible?
Yes
Set $v = v^*$ Calculate err$_p$
Add v to the training set
No
$err_p < \varepsilon_2$
Yes
$p < N_{NLP}$
No
$p = p + 1$
Yes
Update the surrogate model using RBF
No
$\max_{p \in \mathcal{F}} err_p < \varepsilon_2$
Yes
B
Solution Found

Figure 1: Iterative optimization framework

2.4.1 Solution strategy alternatives

The algorithm described above is slightly modified to compare its performance when: a) using only regression algebraic function and b) using only Gaussian Radial Basis Functions. In the first case, i.e. to solve an optimal design problem with surrogate models built exclusively with regression algebraic functions in ALAMO, the exploration (adaptive sampling) and the exploitation (from GDP solutions) are performed using ALAMO in the iterative framework (Fig. 1), and the corresponding surrogate model is updated with new regressions in ALAMO for this step. It is worth noting that the incorporation of a new sampling point during exploration or exploitation steps, implies solving an additional optimization problem to global optimality, including the entire set of sampling points, so the problem becomes more complex as iterations proceed. In the second case; i.e., SM generation with GRBF, the solution strategy of Fig. 1 is also modified, and GRBFs are applied to perform the initial fit. GRBFs coefficients are determined by solving linear systems of equations, and surrogate models are generated through a customized implementation in MATLAB.

2.5 Software integration

In the algorithm depicted in Fig. 1, different software tools are used. We code the main algorithm in MATLAB (R2016a) since this software allows data processing and data transfer to ALAMO (Version 20.10.21) and to GAMS (32.2.0). We also generate data using the Latin Hypercube Sampling (LHS) technique, and solve linear systems of equations to fit customized GRBFs coefficients in MATLAB. We build surrogate models using the learning software ALAMO as it has the capability of selecting low-complexity algebraic basis functions to fit sample data accurately. A ".alm" file is automatically written in MATLAB, containing the possible algebraic basis function and the input-output variables. This file is run in ALAMO through the command windows from MATLAB. The "trace" option is set to generate a summary of the results, from where the function form is read and transferred to MATLAB.

There are four different types of mathematical problems in GAMS, as shown in Fig. 2: 1) Process unit models are formulated and run to generate initial sampling data and to check each process unit solution from the GDP problem; 2) The Error Maximization Sampling is written as an optimization model (Problem 3); 3) The feasibility problem is run in the case when optimal solutions from NLPs subproblems turn out infeasible (Problem 4); 4) The GDP problem is solved to obtain the optimal design (Problem 2). GAMS files (".gms") are opened from MATLAB to write the corresponding function form and its derivatives required for the formulation of Master MILP problems. GDXMRW utilities (Ferris et al., 2011) are employed to write an additional ".gdx" file containing GRBF coefficients, to run mathematical problems in GAMS from MATLAB, and also to upload the most relevant results in MATLAB. The results presented herein were run implementation is run in an Intel(R) Core(TM) i7–4790 CPU @3.60GHz and 8 GB RAM.

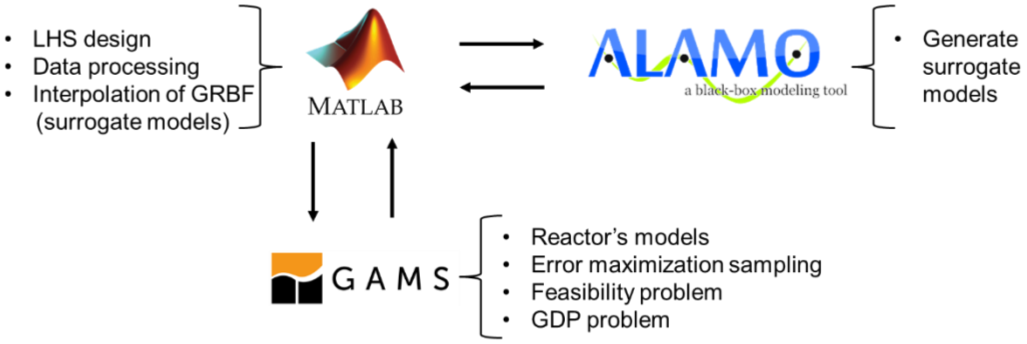

LHS design
Data processing
Interpolation of GRBF (surrogate models)
MATLAB
ALAMO
a black-box modeling tool
Generate surrogate models
GAMS
Reactor's models
Error maximization sampling
Feasibility problem
GDP problem

Figure 2. Software integration

3 CASE STUDY 1: METHANOL SYNTHESIS

3.1 Problem description

The first case study considered to assess the performance of the proposed optimization strategy (Fig. 1) is the methanol synthesis problem described in the literature (Chen and Grossmann, 2019; Türkay and Grossmann, 1996). We consider this simplified methanol synthesis example, which can be solved with state-of-the-art MINLP/GDP solvers. While this optimization model does not intend to provide a rigorous representation of the methanol synthesis reactor, it is considered as a suitable test problem to illustrate the proposed methodology. The process flowsheet is shown in Fig. 3. The main discrete decisions are the selection of the feed stream (1 or 2), which can be conditioned with one (3) or two compression steps (4, 5, and 6). Methanol conversion can be performed in a low-cost and low conversion reactor (9) or in an expensive reactive unit operating with higher methanol production (10). Finally, the recycle stream can be recompressed in one (16) or two steps (17, 18 and 19). The objective function is to maximize plant profit. The detailed equations of the methanol synthesis superstructure model can be found in https://github.com/grossmann-group/gdplib/tree/master/gdplib/methanol or www.logmip.ceride.gov.ar/older.html.

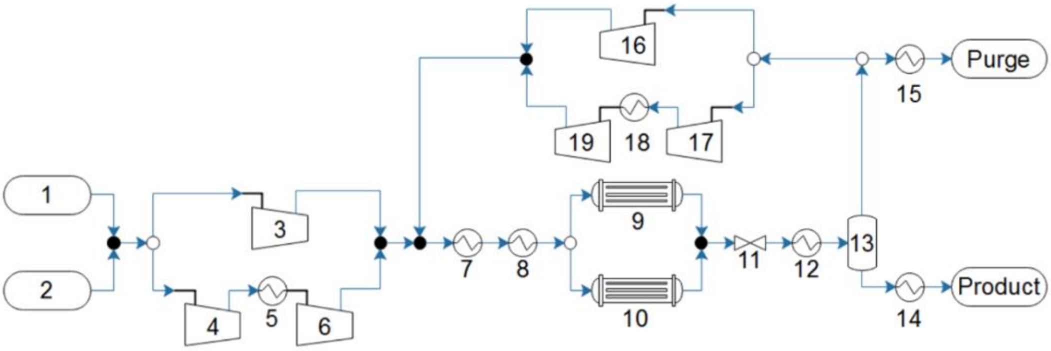


Figure 3. Methanol synthesis superstructure

3.1.1 Reactor model original formulation

We focus on reactor models (9 and 10), which are replaced by surrogate models in the hybrid plant model, to calculate the reaction conversion. The original reactor model (in this paper also referred to as "true model") is as follows:

$$r_u = \chi_u \cdot f_{u,H_2}^{in} \;\; u \in \{9,10\} \tag{5.1}$$

$$\chi_u^{eq} = 0.415\left(1 - \left(\frac{26.25e^{-\frac{18}{T_u^{out}}}}{P_u^{out^2}}\right)\right) u \in \{9,10\} \tag{5.2}$$

$$\chi_9 FT_9^{in} = \chi_9^{eq}(1 - e^{-5})\left(f_{9,H_2}^{in} + f_{9,CO}^{in} + f_{9,CH_3OH}^{in}\right) \tag{5.3}$$

$$\chi_{10} FT_{10}^{in} = \chi_{10}^{eq}(1 - e^{-10})\left(f_{10,H_2}^{in} + f_{10,CO}^{in} + f_{10,CH_3OH}^{in}\right) \tag{5.4}$$

$$(FT_u^{in} T_u^{in} - FT_u^{out} T_u^{out})(35.0) = 0.01\Delta H_{rxn} r_u \;\; u \in \{9,10\} \tag{5.5}$$

$$f_{u,H_2}^{out} = f_{u,H_2}^{in} - r_u \;\; u \in \{9,10\} \tag{5.6}$$

$$f_{u,CO}^{out} = f_{u,CO}^{in} - 0.5 r_u \;\; u \in \{9,10\} \tag{5.7}$$

$$f_{u,CH_3OH}^{out} = f_{u,CH_3OH}^{in} + 0.5 r_u \;\; u \in \{9,10\} \tag{5.8}$$

$$f_{u,CH_4}^{out} = f_{u,CH_4}^{in} \;\; u \in \{9,10\} \tag{5.9}$$

where FT_u^{in} and FT_u^{out} are input and output total flowrates of unit u, respectively; $f_{u,j}^{in}$ and $f_{u,j}^{out}$ are input and output molar flow of component j of unit u, respectively; T_u^{in} and T_u^{out} are input and output stream temperatures of unit u, respectively; P_u^{out} is output stream pressure of unit u; χ_u is the conversion; χ_u^{eq} is the equilibrium conversion; and ΔH_{rxn} is reaction heat (parameter).

3.1.2. Surrogate model for methanol synthesis reactor

In this work, we generate a surrogate model to replace Eqs. (5.1)-(5.4). It calculates the reactor hydrogen consumption (r_u) based on input flow rates (f_{u,H_2}^{in}, $f_{u,CO}^{in}$, f_{u,CH_4}^{in} and f_{u,CH_3OH}^{in}), input temperature (T_u^{in}), and operating pressure (P_u^{in}), as input variables, as follows

$$r_u = h_u^s\left(f_{u,H_2}^{in}, f_{u,CO}^{in}, f_{u,CH_4}^{in}, f_{u,CH_3OH}^{in}, T_u^{in}, P_u^{in}\right) \;\; u \in \{9,10\} \tag{5.10}$$

The hybrid model for reactors is, therefore, formulated with Eqs. (5.5)-(5.10).

3.2 Numerical results for GDP with original model

Numerical results for the original (true) model for the methanol synthesis problem (Eqs. (5.1) – (5.9)) are shown in Table 1. As we employ a custom implementation of the Logic-based Outer

Approximation algorithm, we perform a set covering including two NLPs subproblems to initialize all nonlinear equations for the proper formulation of the Master MILP problem. NLP 1 subproblem includes feed stream 1, two compression steps for this stream (units 4, 5, and 6), reactor 10 and two compression steps (units 17, 18, and 19) for the recycled steam; NLP 2 subproblem considers feeding stream 2, compressor 3, reactor 9, and compressor 16 (see Fig. 3). After solving the first Master MILP problem, the optimal configuration is determined in NLP 3. The optimal profit value is 1,840 M$/y, and the optimal configuration includes feeding stream 2, two compression steps (units 4, 5, and 6), reactor 9, and compressor 16 for the recycle stream. These results are in agreement with those reported in the literature (Türkay and Grossmann, 1996).

Table 1: Logic-based Outer Approximation algorithm iterations (original model)

Iteration/subproblem	Objective (M$/y)	CPU time (s)	Constraints	Continuous vars.	Binary vars.
NLP 1 (CONOPT)	-750.52	0.125	380	310	-
NLP 2 (CONOPT)	1,645.52	0.063	363	310	-
Master MILP 1 (Cplex)	1,156.87	0.140	690	449	17
NLP 3 (CONOPT)	1,840.08	0.063	373	310	-
Master MILP 2 (Cplex)	-2,6734.16	0.125	742	533	17
NLP 4 (CONOPT)	1,194.47	0.079	373	310	-

3.3 Numerical results for GDP with hybrid model

In this section, we solve the optimal design problem constrained with the hybrid model described by Eqs. (5.5)-(5.10) as a GDP within the novel algorithm proposed in this work and its alternatives. Due to the stochastic component of LHS technique in the proposed solution strategy (Fig. 1), the case study is solved ten times using different initial sampling points (obtained from LHS) for testing each analyzed technique. In all cases, the optimal solution process scheme is obtained with slight variations in the objective function, lower than 0.24 %. The comparison is usually focused on the number of major iterations and CPU time devoted to each main step in the proposed algorithm.

3.3.1 Surrogate model generated with algebraic regression functions in ALAMO

Figure 4 shows optimization results for ten different LHS initial points for the methanol synthesis case study when SM are generated with simple algebraic regression functions in ALAMO. The number of major iterations is between 1 and 4. When the problem is solved in one iteration (case 2), the SM after the adaptive sampling step is so accurate that it satisfies the convergence criteria without requiring an exploitation step (i.e., further refinement). Regarding the objective function value, numerical results show that this method slightly overestimates the plant profit in every case, being the largest overestimation of 0.22% (the mean value for the objective function is 1,842.4 M$/y). CPU times are also shown in Fig. 4, and they are broken down in the main steps: initial fit, true simulation optimal value, adaptive sampling, and GDP problem solution. CPU time varies between 18 and 77 s, and its mean is 42 s. The most time-consuming step corresponds to adaptive sampling (around 70 % of total CPU time), which requires an average of 11 iterations to satisfy the accuracy criteria. This result includes solving NLPs of Error Maximization Sampling problems and updating the model using ALAMO, which involves solving additional optimization problems (as explained in Section 2.4.1). It is also observed that CPU time for the initial fit is negligible (1-2 s). Explicit equations obtained with ALAMO for the last run in Fig. 4 are provided in the Supplementary Material.

An alternative approach would be to generate a one-shot high accuracy SM with ALAMO. However, some functions in the original model are highly nonlinear and cannot be appropriately described by using simple algebraic regression functions. Moreover, it is difficult to know beforehand the accuracy of the generated model in the neighborhood of the optimal solution. Furthermore, the main feature of ALAMO is to generate simple functions by finding a trade-off solution between the model complexity and model errors. For that reason, in Section 3.3.3, we propose to generate an initial simple SM using ALAMO, which is further refined using GRBFs.

We also analyze the influence of the initial SM generated with ALAMO in the algorithm performance, considering that the accuracy of the initial SM increases with the number of sampling points. Thus, we solve the case studies with different numbers of initial sampling points in Section 3.3.3. Even in the runs with a large number of initial sampling points, the algorithm requires the exploration and the exploitation steps to ensure an accurate optimal solution.

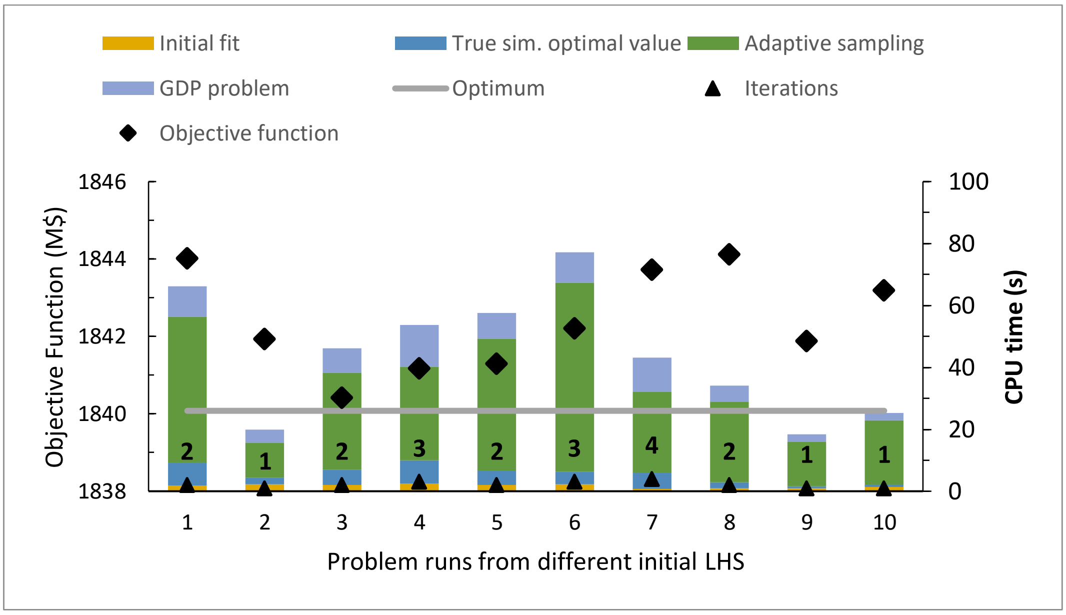


Figure 4: Methanol synthesis problem constrained with hybrid model using surrogate models based on simple algebraic regression functions

3.3.2 Surrogate model generated with Gaussian Radial Basis Functions

Figure 5 shows numerical results corresponding to the methanol synthesis case study, when only GRBF-based surrogate models are generated. We solve the problem ten times under different LHS designs. In 60 % of the cases, three major iterations are required to satisfy the convergence criteria. The objective function is between 1,836 and 1,842 M$/y, which corresponds to both underestimation and overestimation of the profit relative to the true value (1840 M$), and the mean is 1,840.0 M$/y. The relative error is 0.24 % in the worst case.

Regarding CPU time distribution, Fig. 5 also shows the corresponding breaking down. Total CPU time is between 13 and 35 s, which suggests that this solution strategy may be more time efficient than the one described in 3.3.1. The initial fit time is not distinguishable in Fig. 5, as it requires solving a linear system of equations. The adaptive sampling is the most time-consuming step, representing 61 % of total CPU time, in average, and requiring 18 iterations in average. However, CPU time is lower than the time required in the case of using algebraic regression functions for surrogate model formulation (see Fig. 4 and Section 3.3.1), as the interpolating GRBF updating is carried out by solving a linear system of equations in MATLAB.

Furthermore, it can be seen that CPU time associated with obtaining the solution of the GDP problem does increase with the number of major iterations, as a GDP is solved in each major iteration.

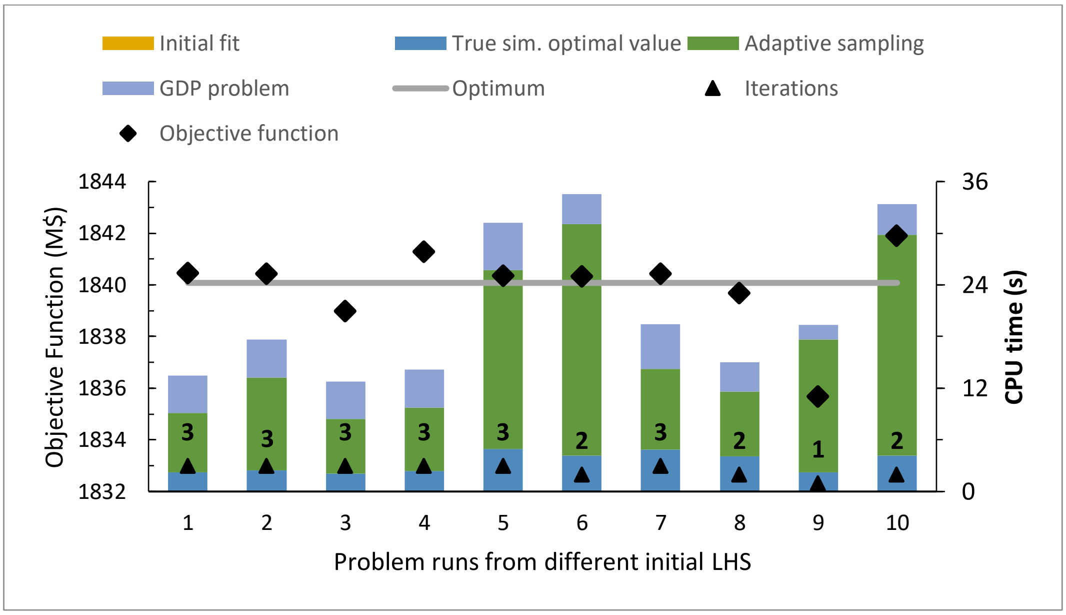


Figure 5: Methanol synthesis problem solved constrained with hybrid model using surrogate models based on GRBFs

3.3.3 Surrogate model generation with algebraic regression and Gaussian Radial Basis Functions (ALAMO and customized GRBF)

Figure 6 shows numerical results corresponding to the case of using the iteration framework described in Fig. 1 to solve the methanol synthesis problem with surrogate models represented by Eq. (1). It shows that the number of major iterations is between two and four. The relative error of the objective function is 0.11 % in the worst case, suggesting that this is the most robust strategy to address this problem. It is also the most efficient when compared to the two previous strategies, requiring roughly 16 s on average per problem run; which means CPU time savings of 26 % and 49 %, as compared to using only GRBFs or algebraic regression functions, respectively. This result is mainly associated with CPU time savings in the adaptive sampling step. As the initial fit is performed using algebraic regression functions, the number of iterations in the adaptive sampling step (16 as average) is often lower than in the case of using GRBF-based surrogate models. Furthermore, due to the function complexity in process design models, ALAMO can usually capture the main function patterns, and in this way, the required number of interpolation points is reduced, as compared to the case of GRBF-based SMs. In the "combined strategy" (Fig. 1), the initial fit is performed through simple algebraic regression functions and GRBF are only added in the exploration and exploitation steps. In this case, we also provide the explicit equations obtained with ALAMO as initial SM for the last run of Fig. 6 in the Supplementary Material.

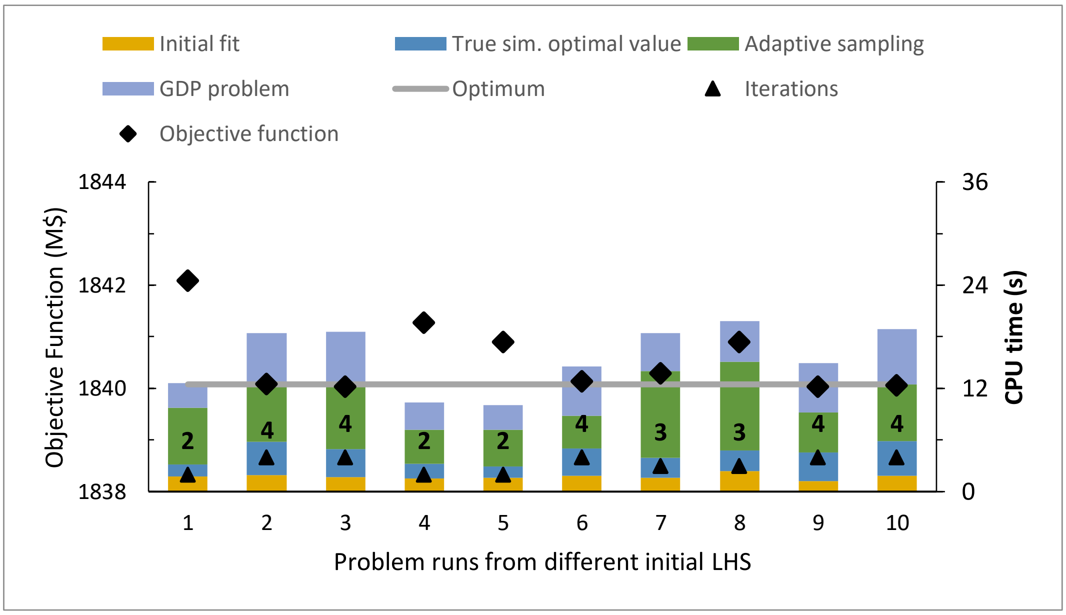


Figure 6: Methanol synthesis problem constrained with hybrid models using surrogate models based on simple algebraic regression functions and GRBFs

In order to analyze the impact of the number of sampling points on the initial surrogate model, the case study is solved using different sampling data sets obtained with the LHS approach. Furthermore, as this sampling technique has a random component, the case study is solved ten times in each case to analyze the performance of the proposed method. This analysis is carried out considering SM based on both regression functions and GRBFs, using the solution strategy shown in Fig. 1. Numerical results related to 100 initial sampling points (shown in Fig. 6) are considered the base case for comparison purposes.

Figure 7 shows numerical results from the methanol case using 10 initial sampling points (see Fig. 1). It should be noted that this number of points corresponds to the initial LHS, and some of them could be discarded in the filtering step. As in these cases, the number of points is small and the objective function is the Bayesian Information Criterion (BIC); the initial SM from ALAMO generates interpolation functions. It is observed that the number of major iterations increases when the number of initial sampling points is reduced. For example, in two runs with 10 initial sampling points, eight iterations were required, while four iterations were necessary in the worst runs for the base case (i.e., with 100 initial points).

Consequently, the mean CPU time increases 38 % with respect to the base case. Furthermore, it is observed that CPU times are reduced by 50 % as related to the initial fit, compared to the case of 100 initial sampling points (0.9 s vs. 1.8 s on average). On the other hand, CPU time related to the adaptive sampling step shows a 51 % increase (9.8 s vs. 6.5 s on average) since more exploration iterations are required to generate a model with the appropriate accuracy in the entire domain, as compared to the base case.

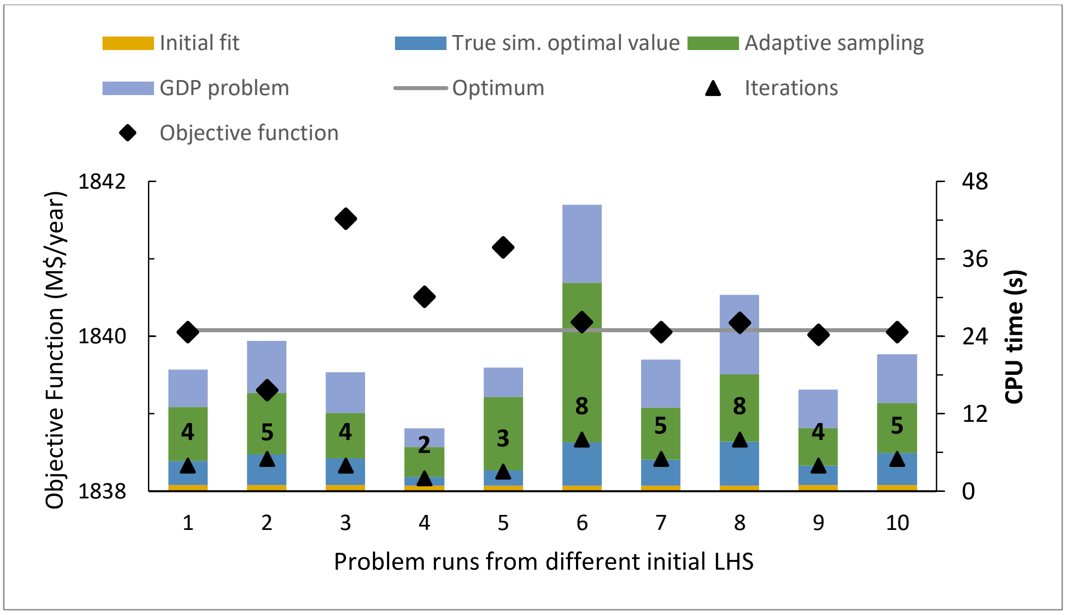


Figure 7. Methanol synthesis problem constrained with hybrid models using surrogate models based on simple algebraic regression functions and GRBFs (10 initial sampling points)

The methanol synthesis problem is also solved using 1000 initial sampling points, and the corresponding results are shown in Fig. 8. It can be noted that in 50 % of the runs, two iterations are required to solve the problem when using 1000 initial sampling points, and five iterations are necessary in the worst case (two times), whereas four iterations are carried out in the worst run for the base case with 100 initial points (see Fig. 6). As a result, the average CPU time is marginally reduced by 6.3 % (15 s vs. 16 s). On the one hand, CPU time associated with the initial fit increases by 57 % (2.8 vs. 1.8 s), and this step of the algorithm represents 21 % of total CPU time on average. On the other hand, there is a decrease of 29 % in computational time associated with adaptive sampling, regarding the base case (4.6 vs. 6.5 s in average). This fact clearly shows that the larger the number of initial sampling points, the smaller is the number of iterations in the exploration stage.

Regarding the objective function, it is observed that all approximations obtained with a different number of initial sampling points have errors lower than 1 % with respect to the true optimal solution. Furthermore, it should be pointed out that the same optimal scheme is determined in every case, which is in agreement with the one obtained with the true GDP model. These results indicate the efficiency of the exploration and exploitation stages to refine the SM.

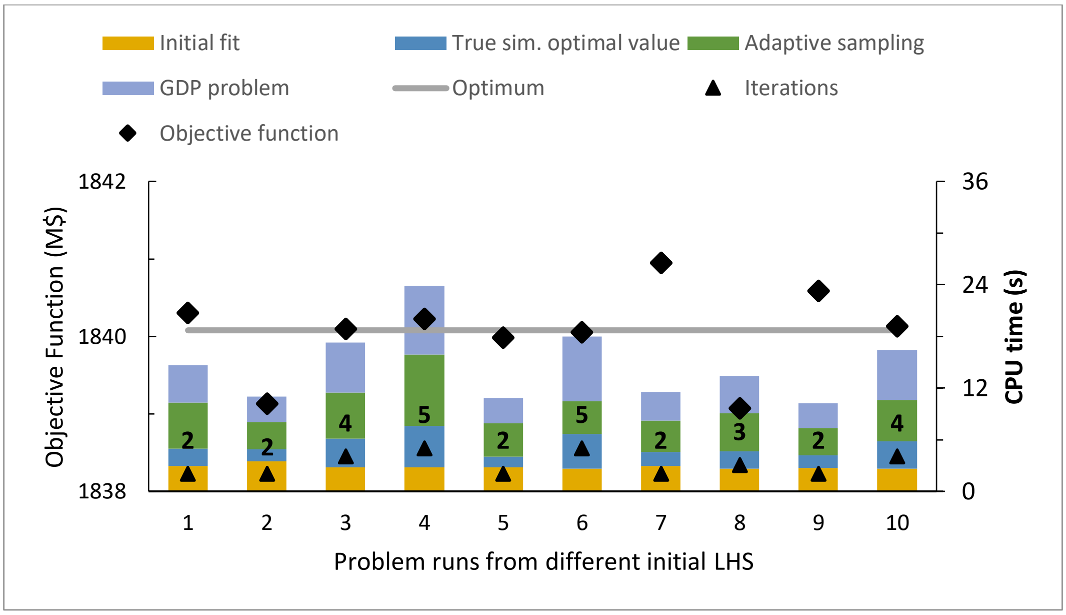


Figure 8. Methanol synthesis problem constrained with hybrid models using surrogate models based on simple algebraic regression functions and GRBFs (1000 initial sampling points)

3.3.3.1 Numerical results for GDP constrained with true model and with hybrid model

To compare numerical results for the methanol synthesis problem by solving a GDP constrained with the original (true) model and the ones obtained with the proposed algorithm for hybrid model generation, we consider the last (fourth) GDP results in case 10 from Fig. 6.

Table 2 shows objective function values and the number of variables in each iteration. The same set covering described in Section 3.2 is used to define NLP 1 and NLP 2 for the methanol synthesis case study (Fig. 3). As with the true model, the optimal scheme is determined in NLP 3 (see Section 3.2). The optimal scheme found is the same as the one obtained with the original model. Furthermore, the objective function value obtained with the proposed procedure has a relative error of less than 0.002% with respect to the true solution (1,840.05 vs. 1,840.08 M$/y). Additionally, and due to the refinement step of NLPs subproblems, relative errors of NLP 1 and NLP 2 (when compared to the true model solution) are 0.02% and 0.04%, respectively.

The proposed methodology does not show any advantages in this case study due to its low complexity. As it can be seen in Tables 1 and 2 and Fig. 6, the CPU time required for the true model solution is lower than the required when using the hybrid model (96 %). These results are mainly associated with the exploration and exploitation steps to refine the SM. However, applying the proposed methodology to this case study is worthwhile since it helps to illustrate the methodology and its robustness. In Section 4, we apply the solution strategy to the design of a propylene production plant via olefins metathesis, where the methodology advantages are clearly shown.

Table 2. Logic-based Outer Approximation algorithm iterations (hybrid model) for the 4th GDP problem in case 10 (Fig. 6)

Iteration/subproblem	Objective (M$/y)	CPU time (s)	Constraints	Continuous vars.	Binary vars.
NLP 1 (CONOPT)	-750.70	0.078	378	310	-
NLP 2 (CONOPT)	1,646.17	0.047	361	310	-
Master MILP 1 (Cplex)	1,159.21	0.265	682	451	17
NLP 3 (CONOPT)	1,840.05	0.031	371	310	-
Master MILP 2 (Cplex)	428.62	0.156	730	521	17
NLP 4 (CONOPT)	1,540.76	0.047	368	310	-

Regarding optimal values for input and output variables for reactor 9, a comparison is shown in Table 3 for the different basis functions to generate surrogate models, and Table 4 shows the corresponding relative differences with respect to the true model. It can be noted that relative errors are negligible in input variables since the highest one is 0.04 % for the methane input flowrate with respect to the obtained with the GDP constrained with the original model. For the output variable, the relative error is higher than the latter case, but it is still negligible (0.10 %).

Table 3. Comparison of optimal values for input and output variables in reactor 9 surrogate model

SM Input variables	Original model	ALAMO SM	GRBF SM	Proposed SM
$f_{9,H_2}^{in}(kmol/s)$	6.789	6.782	6.803	6.789
$f_{9,CO}^{in}(kmol/s)$	1.034	1.022	1.034	1.034
$f_{9,CH_3OH}^{in}(kmol/s)$	4.486	4.488	4.500	4.487
$f_{9,CH_4}^{in}(kmol/s)$	4.463	4.481	4.476	4.465
$T_9^{in}(100K)$	4.553	4.556	4.553	4.553
$P_9^{in}(MPa)$	13.817	13.852	13.817	13.819
SM Output variables				
$r_9(kmol/s)$	2.043	2.043	2.045	2.045

Table 4. Relative errors of optimal values for input and output variables in reactor 9 surrogate model, with respect to the original model

SM Input variables	ALAMO SM	GRBF SM	Proposed SM
$f_{9,H_2}^{in}(kmol/s)$	0.10%	0.21%	0.00%
$f_{9,CO}^{in}(kmol/s)$	1.16%	0.00%	0.00%
$f_{9,CH_3OH}^{in}(kmol/s)$	0.04%	0.31%	0.02%
$f_{9,CH_4}^{in}(kmol/s)$	0.40%	0.29%	0.04%
$T_9^{in}(100K)$	0.07%	0.00%	0.00%
$P_9^{in}(MPa)$	0.25%	0.00%	0.01%
SM Output variables			
$r_9(kmol/s)$	0.00%	0.10%	0.10%

3.3.4 Contribution of simple algebraic regression and Gaussian Radial Basis functions

In the initial surrogate model for reactor 9 hydrogen consumption of the methanol synthesis example, one hundred sampling points were generated via LHS. For these data, ninety simulations were feasible, so ten points were discarded due to infeasible simulations. ALAMO was used to build a simple algebraic regression surrogate model using the basis functions detailed in Section 2.1, including 44 terms. Sampling point relative errors are shown in Fig. 9, where a value of 3 % is used for tolerance ε_0. It is observed that four sampling points have relative errors larger than the given tolerance, so these points are included in the set of interpolation points for the Gaussian Radial Basis Function (generation of four GRBFs).

It should be noted that the tolerance is not always achievable using only algebraic regression functions. On the other hand, addressing the problem only through interpolating functions results in a complex function for ninety interpolating points in this case. The proposed approach allows satisfying the given tolerance with a combination of basis functions that is simpler and may be more appropriate for optimization than using only interpolating functions.

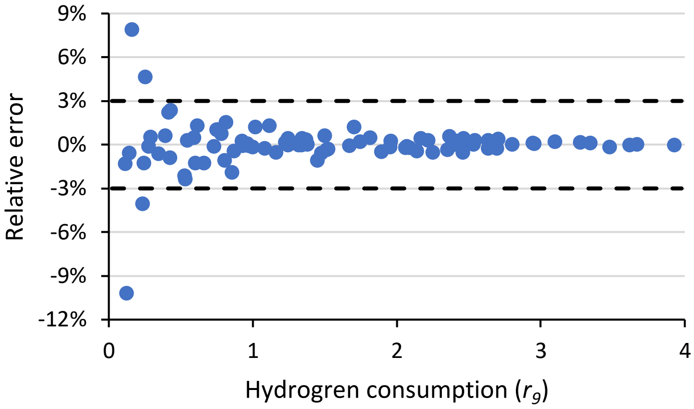


Figure 9: Relative errors of sampling points for reactor 9 surrogate model using algebraic regression functions

In order to assess the weight of each group of basis functions in the surrogate model generated with Eq.(1), we build 3D plots for reactor 9 surrogate model, which calculates hydrogen consumption, at the end of a problem run. In particular, input variables component molar flows are fixed at the values shown in Table 3, while temperature and pressure are considered as variables. Figures 10a and 10b show the contribution of simple algebraic regression functions and GRBF to the output variable, respectively. Figure 10b shows that the maximum absolute value of GRBF is less than 2% of the value calculated with simple algebraic regression functions. Figure 11 shows the form of the proposed surrogate model, which is the sum of simple algebraic functions (Fig. 10a) and GRBF (Fig. 10b). It is observed that the form of the surrogate from ALAMO is retained, while GRBF refine special domain parts where the accuracy is not large enough as seen in Figs. 10 and 11.

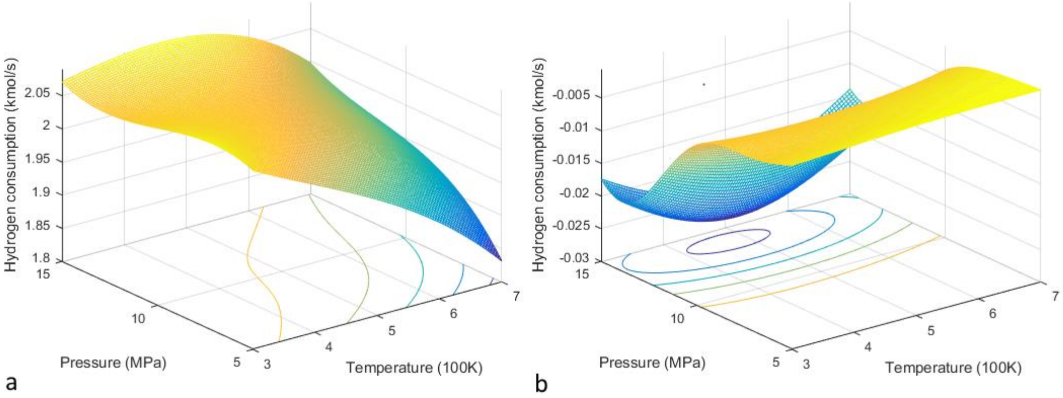


Figure 10: Surrogate model for reactor 9. a) contribution of simple algebraic regression functions in ALAMO. b) contribution of GRBF after exploration and exploitation steps

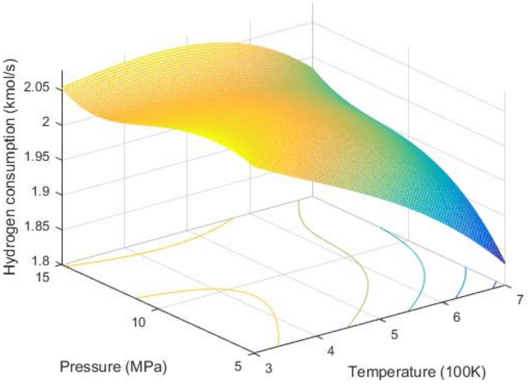


Figure 11: Surrogate model for reactor 9 based on simple regression algebraic functions and GRBF

4 CASE STUDY 2: PROPYLENE PRODUCTION VIA OLEFIN METATHESIS

4.1 Problem description

As a second case study, we consider the optimal design of a propylene production plant via olefin metathesis, whose superstructure is shown in Fig. 12. The available feed streams are ethylene and a butene mix. Compressors C1, C2, and C3 are conditional units that can be included in the optimal design. A fraction of the ethylene stream can be optionally sent to a dimerization process (RDIM) to produce 1-butene. The rest of the ethylene stream is sent to the metathesis reactor. The butene stream is fed to a hydrogenation reactor unit (C4H) to convert butadiene and ethyl-acetylene into butenes. Further, the butene stream is processed in an isomerization reactor (ISOM) to increase the trans-2-butene content. The isomerization reactor output stream is fed to the metathesis reactor (MTR), where propylene production takes place. The olefin mixture is purified in a separation train to obtain ethylene, propylene, and C_4^+. As the optimal distillation column sequence is unknown beforehand; we model its superstructure through a State Equipment Network (SEN) representation to formulate the task assignment in equipment units, and consequently, the separation step order (Mencarelli et al., 2020; Pedrozo et al., 2020; Yeomans and Grossmann, 1999). We include the deethylenizer (DC1) and the depropylenizer (DC2) in the SEN superstructure. The main product is propylene, while ethylene, obtained from the deethylenizer top stream, is recycled. Additionally, the depropylenizer bottom stream has C5 and C6 olefins, which are separated in DC3 column. The bottom stream of this column is sold as a byproduct (C_5^+), while the top stream is recycled. Butane can be highly concentrated in the DC3 top stream since its concentration in the butene input stream is 10 %, and it is inert in the process. Therefore, a fraction of DC3 top stream is sold as a byproduct (butane). The objective function is the net present value (NPV) maximization. A detailed description of the entire plant model is presented in Pedrozo et al. (Pedrozo et al., 2021) and the SEN model can be found in the Supplementary Material. In this work, we focus on the distillation train, in which rigorous column models are replaced by surrogate models. Raw material, utility costs and product prices are shown in Table 5 (Boulamanti and Moya, 2017).

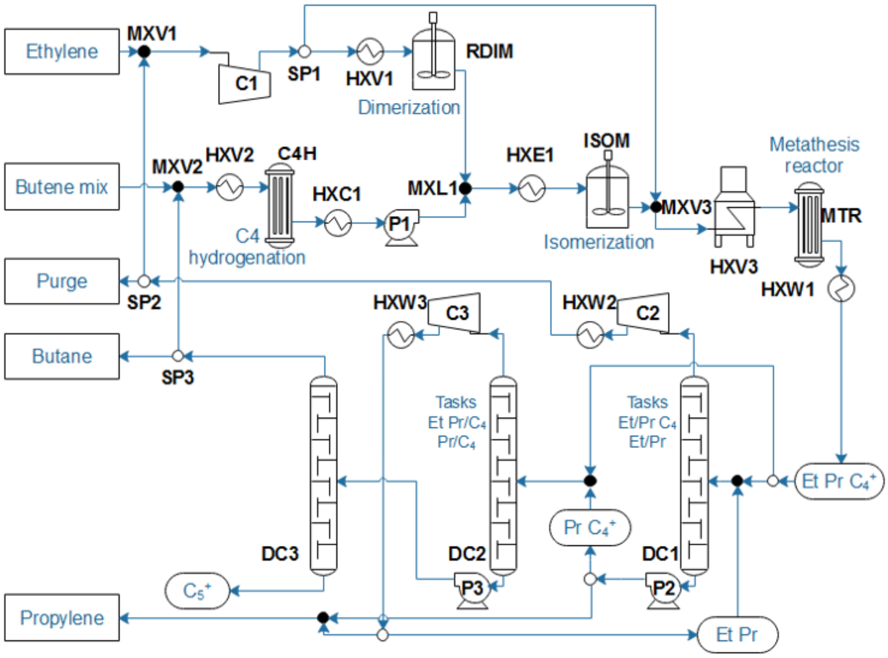


Figure 12: Olefin metathesis process superstructure

Table 5: Raw material, utility costs and product prices (Boulamanti and Moya, 2017)

	Units	USA
Natural gas	\$/t	174
Ethylene	\$/t	1,138
Butenes	\$/t	978
Hydrogen	\$/t	429
Electricity	\$/MWh	41
Propylene	\$/t	1,205
Butane	\$/t	550
C_5^+	\$/t	906

4.2 Distillation column train

4.2.1 Rigorous distillation column model

We formulate rigorous MESH (mass-equilibrium-summation-enthalpy) models for distillation column units (Biegler et al., 1997; Viswanathan and Grossmann, 1993). Thus, we obtain the “original or true” distillation columns models in this work. Feeding stream specifications are described in Eqs. (6.1) and (6.2). Material balances are expressed through Eqs. (6.3)-(6.6). The thermodynamic equilibrium condition is formulated in (6.7) and (6.8) considering ideal gas and Raoult’s law. Summation equations are (6.9) and (6.10) for liquid and vapor stream, respectively, while enthalpy balances are in Eqs. (6.11)-(6.16).

A linear pressure profile and a fixed pressure drop of 1 bar is set by Eqs. (6.17) and (6.18); and a temperature profile is constrained in (6.19). Pure component properties are calculated using Eqs. (6.20)-(6.22), where partial pressure is calculated with the extended Antoine equation (Green and Perry, 2007). Eqs. (6.23)-(6.30) are connecting equations. Operating costs due to condenser and reboiler duty are calculated using correlations proposed by Ulrich and Vasudevan (Ulrich and Vasudevan, 2006) in Eqs. (6.31) and (6.32). Capital cost is estimated using Eq. (6.33).

$$f_{u,j}^{in} = DF_{u,n,j} \qquad j\epsilon J, n\epsilon N_u^{Feed}, u\epsilon Udc \tag{6.1}$$

$$H_u^{in} = DH_{u,n} \qquad (u1,n)\epsilon NFe_u, u\epsilon Udc \tag{6.2}$$

$$DF_{u,n,j} + Dfl_{u,n-1,j} + Dfv_{u,n+1,j} = Dfl_{u,n,j} + Dfv_{u,n,j} \quad j\epsilon J, n\epsilon N_u^{Feed}, u\epsilon Udc \tag{6.3}$$

$$Dfl_{u,n-1,j} + Dfv_{u,n+1,j} = Dfl_{u,n,j} + Dfv_{u,n,j} \quad j\epsilon J, n\epsilon NS_u \backslash N_u^{Feed}, u\epsilon Udc \tag{6.4}$$

$$Dfv_{u,n+1,j} = Dfl_{u,n,j} + Dfv_{u,n,j} \qquad j\epsilon J, n\epsilon CD_u, u\epsilon Udc \tag{6.5}$$

$$Dfl_{u,n-1,j} = Dfl_{u,n,j} + Dfv_{u,n,j} \qquad j\epsilon J, n\epsilon RB_u,, u\epsilon Udc \tag{6.6}$$

$$DL_{u,n} = DV_{u,n}DR_{u,n} \;\; n\epsilon\{NS_u \cup CD_u \cup RB_u\}, u\epsilon Udc \tag{6.7}$$

$$Dfv_{u,n,j}DR_{u,n}DCp_{u,n} = Dfl_{u,n,j}\exp\left(DLp_{u,n,j}\right) \;\; j\epsilon J, n\epsilon\{NS_u \cup CD_u \cup RB_u\}, u\epsilon Udc \tag{6.8}$$

$$\sum_{j\epsilon J} Dfl_{u,n,j} = DL_{u,n} \;\; n\epsilon\{NS_u \cup CD_u \cup RB_u\}, u\epsilon Udc \tag{6.9}$$

$$\sum_{j\epsilon J} Dfv_{u,n,j} = DV_{u,n} \;\; n\epsilon\{NS_u \cup CD_u \cup RB_u\}, u\epsilon Udc \tag{6.10}$$

$$DH_{u,n} + Dhl_{u,n-1} + Dhv_{u,n+1} = Dhl_{u,n} + Dhv_{u,n} \;\; j\epsilon J, n\epsilon N_u^{Feed}, u\epsilon Udc \tag{6.11}$$

$$Dhl_{u,n-1} + Dhv_{u,n+1} = Dhl_{u,n} + Dhv_{u,n} \;\; j\epsilon J, n\epsilon NS_u \backslash N_u^{Feed}, u\epsilon Udc \tag{6.12}$$

$$Dhv_{u,n+1} - Q_u^{cond} = Dhl_{u,n} + Dhv_{u,n} \;\; j\epsilon J, n\epsilon CD_u, u\epsilon Udc \tag{6.13}$$

$$Dhl_{u,n-1} + Q_u^{reb} = Dhl_{u,n} + Dhv_{u,n} \;\; j\epsilon J, n\epsilon RB_u, u\epsilon Udc \tag{6.14}$$

$$Dhv_{u,n} = \sum_{j\epsilon J} Dhvc_{u,n,j}Dfv_{u,n,j} \;\; n\epsilon\{NS_u \cup CD_u \cup RB_u\}, u\epsilon Udc \tag{6.15}$$

$$Dhl_{u,n} = \sum_{j\epsilon J} Dhlc_{u,n,j}Dfl_{u,n,j} \;\; n\epsilon\{NS_u \cup CD_u \cup RB_u\}, u\epsilon Udc \tag{6.16}$$

$$P_u^{top} + 1 = P_u^{bottom} \;\; u\epsilon Udc \tag{6.17}$$

$$DCp_{u,n-1} - 2\,DCp_{u,n} + DCp_{u,n+1} = 0 \;\; j\epsilon J, n\epsilon NS_u, u\epsilon Udc \tag{6.18}$$

$$DCt_{u,n} \leq DCt_{u,n-1} \;\; n\epsilon NS_u, u\epsilon Udc \tag{6.19}$$

$$Dhvc_{u,n,j} = H_j^0 + \sum_{i=1}^{4} \frac{c_{j,i}^{CP}}{i}\left({DCt_{u,n}}^{i} - \left(T^{ref}\right)^i\right) \;\; j\epsilon J, n\epsilon\{NS_u \cup CD_u \cup RB_u\}, u\epsilon Udc \tag{6.20}$$

$$Dhlc_{u,n,j} = \sum_{i=1}^{5} c_{j,i}^{HL}{DCt_{u,n}}^{i-1} \;\; j\epsilon J, n\epsilon\{NS_u \cup CD_u \cup RB_u\}, u\epsilon Udc \tag{6.21}$$

$$DLp_{u,n,j} = c_{j,1}^{AN} + \frac{c_{j,2}^{AN}}{DCt_{u,n}} + c_{j,3}^{AN}Log\left(DCt_{u,n}\right) + c_{j,4}^{AN}{DCt_{u,n}}^{c_{j,4}^{AN}} \;\; j\epsilon J, n\epsilon\{NS_u \cup CD_u \cup RB_u\}, u\epsilon Udc \tag{6.22}$$

$$Dfl_{u,n,j} = f_{u,j}^{bottom} \;\; j\epsilon J, n\epsilon RB_u, u\epsilon Udc \tag{6.23}$$

$$Dfv_{u,n,j} = f_{u,j}^{top} \; j\epsilon J, n\epsilon CD_u, u\epsilon Udc \tag{6.24}$$

$$Dhl_{u,n} = H_u^{bottom} \; n\epsilon RB_u, u\epsilon Udc \tag{6.25}$$

$$Dhv_{u,n} = H_u^{top} \; n\epsilon CD_u, u\epsilon Udc \tag{6.26}$$

$$DCp_{u,n} = P_u^{bottom} \; n\epsilon RB_u, u\epsilon Udc \tag{6.27}$$

$$DCp_{u,n} = P_u^{top} \; n\epsilon CD_u, u\epsilon Udc \tag{6.28}$$

$$DCt_{u,n} = T_u^{top} \; n\epsilon RB_u, u\epsilon Udc \tag{6.29}$$

$$DCt_{u,n} = T_u^{bottom} \; n\epsilon CD_u, u\epsilon Udc \tag{6.30}$$

$$CcUt_u = f^{cu}\left(T_u^{top}, Q_u^{cond}\right) \;\; n\epsilon CD_u, u\epsilon Udc \tag{6.31}$$

$$ChUt_u = f^{hu}\left(T_u^{bottom}, Q_u^{reb}\right) \;\; n\epsilon RB_u, u\epsilon Udc \tag{6.32}$$

$$CC_u = g^{DC}\left(Dfl_{u,n,j}, Dfv_{u,n,j}, DCp_{u,n}, DCt_{u,n}\right), u\epsilon Udc \tag{6.33}$$

where $DF_{u1,u,n,j}$ are input flowrates from unit $u1$ to column u in stage n of component j; $DH_{u1,u,n}$ is input enthalpy flow from unit $u1$ to column u in stage n; $Dfv_{u,n,j}$ and $Dfl_{u,n,j}$ are the vapor and liquid molar flow of component j at stage n in column u, respectively; $DV_{u,n}$ and $DL_{u,n}$ are vapor and liquid total molar flow at stage n in column u, respectively; $DR_{u,n}$ is the liquid-to-vapor ratio at stage n; $DCt_{u,n}$ and $DCp_{u,n}$ are the temperature and pressure at stage n, respectively; $DLp_{u,n,j}$ is the logarithm of pure component j partial pressure at stage n; $Dhl_{u,n}$ and $Dhv_{u,n}$ liquid and vapor enthalpy flow from stage n, respectively; $Dhvc_{u,n,j}$ and $Dhlc_{u,n,j}$ are vapor and liquid enthalpy of pure component j from stage n, respectively; Q_u^{reb} and Q_u^{cond} are the reboiler and condenser duty in column u; $CcUt_u$ and $ChUt_u$ are cooling and heating utility of unit u, and CC_u is the capital cost of of unit u. N_u^{Feed} is the feed stage in column u; NS_u are separation stages in distillation column u; CD_u and RB_u are condenser and reboiler stage in unit u, respectively; Udc is the set of columns; J is the set of components.

4.2.2. Hybrid model for distillation column

In this work, we generate a hybrid model to describe the distillation column performance. Although shortcut models, such as Fenske-Underwood-Gilliland (FUG), are popular choices to represent distillation columns in order to simplify the optimization model, they may also provide inaccurate estimations, for instance, associated with condenser and reboiler heat duties (Dowling and Biegler, 2015). We found relative errors higher than 10 % in these duties when shortcut distillation column models are used for multicomponent separation processes. On the other hand, surrogate models can be refined to achieve a desirable accuracy (i.e., less than 1 %, in this work) as compared to a rigorous model.

Input variables for the surrogate model are component molar flowrates and specific enthalpy for feed stream and column top pressure. Surrogate models are generated with Eq. (1) to calculate equipment capital cost, reboiler heat duty, top and bottom temperatures. These models are shown in Eqs. (7.1)-(7.4) (i.e., the hybrid model replaces Eqs. (6.1)-(6.33) by (7.1)-(7.8), (6.17), (6-31)-(6.32)). We assume sharp separation and fixed pressure drop of 1 bar in the columns. Furthermore, Eq. (7.5) defines the stream specific enthalpy (Hs_u) and Eq. (7.6) is an energy balance to determine condenser duty. Eqs. (7.7) and (7.8) are top and bottom stream enthalpy calculation, respectively, from output flowrates and the corresponding temperatures.

$$CC_u = h_1^s\left(f_{u,j}^{in}, Hs_u, P_u^{top}\right) \; u\epsilon Udc \tag{7.1}$$

$$Q_u^{reb} = h_2^s\left(f_{u,j}^{in}, Hs_u, P_u^{top}\right) \; u\epsilon Udc \tag{7.2}$$

$$T_u^{top} = h_3^s\left(f_{u,j}^{in}, Hs_u, P_u^{top}\right) \; u\epsilon Udc \tag{7.3}$$

$$T_u^{bottom} = h_4^s\left(f_{u,j}^{in}, Hs_u, P_u^{top}\right)\ u\epsilon Udc \qquad (7.4)$$

$$Hs_u = 10^3\ H_u^{in}/\textstyle\sum_{j\in J} f_{u,j}^{in}\ \ u\epsilon Udc \qquad (7.5)$$

$$H_u^{in} - Q_u^{cond} + Q_u^{reb} = H_u^{top} + H_u^{bottom}\ u\epsilon Udc \qquad (7.6)$$

$$H_u^{top} = \textstyle\sum_{j\in J} f_{u,j}^{top}\left(H_j^0 + \sum_{i=1}^4 \frac{c_{j,i}^{CP}}{i}\left(\left(T_u^{top}\right)^i - (T^{ref})^i\right)\right)\ u\epsilon Udc \qquad (7.7)$$

$$H_u^{bottom} = \textstyle\sum_{j\in J} f_{u,j}^{bottom}\left(\sum_{i=1}^5 c_{j,i}^{HL}(T_u^{bottom})^{i-1}\right)\quad u\epsilon Udc \qquad (7.8)$$

4.3 Optimal propylene plant design by solving GDP constrained with rigorous model

In this case, it is possible to select an initial process configuration that includes all equipment units. Therefore, one NLP subproblem (NLP 1) is enough to initialize all nonlinear equations for the formulation of the first Master MILP problem. Numerical results and model statistics of NLP and MILP subproblems are shown in Table 6 for the rigorous model (Eqs. (6.1) – (6.33)) (in this paper also referred to as “true model”). An optimal NPV value of 126 MM$ is found in NLP 2 in roughly 176 s of total CPU time. NLP subproblems have around 6300 constraints.

The optimal scheme is shown in Fig. 13. Both feed steams are selected, ethylene and butenes mix. In particular, ethylene dimerization is not selected within the optimal scheme, probably due to the high ethylene price. Only compressor C2 is selected for DC1 top stream. Regarding the separation scheme, the depropylenizer column (DC2) is selected as the first one, and the deethylenizer (DC1) is fed with DC2 top stream. This separation train order may be due to the high output temperature of the metathesis reactor (506 K).

Table 6: Logic-based Outer Approximation algorithm iterations for propylene production plant with rigorous model (Eqs. (6.1) – (6.33))

Iteration/subproblem	Objective (MM$)	CPU time (s)	Constraints	Continuous vars.	Binary vars.
NLP 1 (CONOPT)	85.63	6.56	6,547	6,303	-
Master MILP 1 (Cplex)	170.67	8.23	12,615	14,758	34
NLP 2 (CONOPT)	125.77	60.00	6,395	6,154	-
Master MILP 2 (Cplex)	146.68	54.14	20,644	22,861	34
NLP 3 (CONOPT)	123.34	46.73	6,390	6,148	-

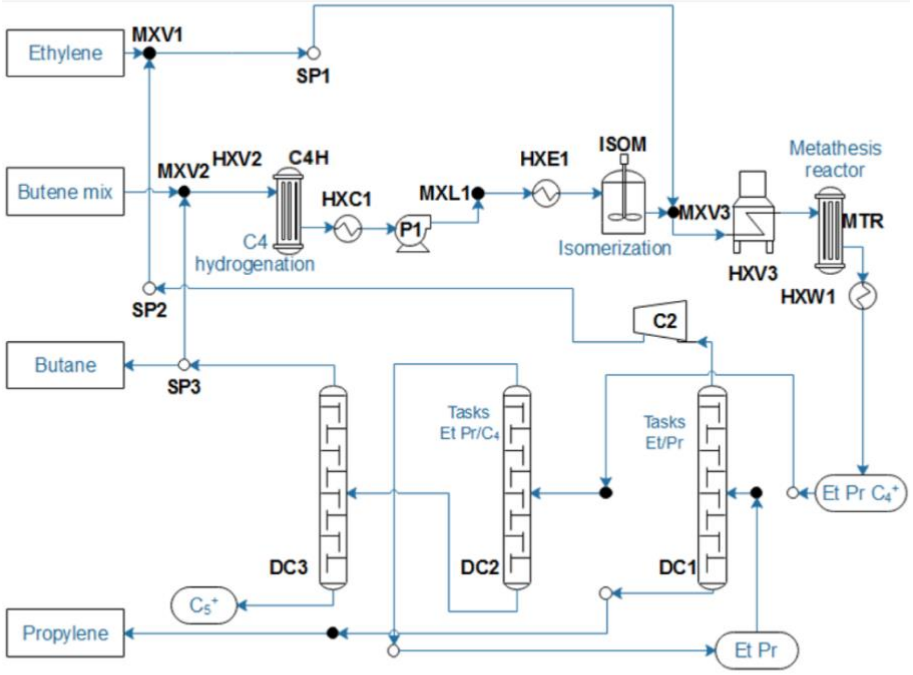


Figure 13: Optimal scheme for propylene production via olefin metathesis

4.4 Optimal propylene plant design by solving GDP constrained with hybrid model

4.4.1. Hybrid model performance with different initial LHS designs

The optimal design problem for a propylene production plant via olefin metathesis is solved ten times under different initial LHS designs. Figure 14 shows the corresponding results and the optimal scheme is the same as the one determined with the original model in all cases, which highlights the robustness of the proposed solution strategy. The algorithm requires three major iterations (solving three GDPs) to satisfy the convergence criteria in four cases and, in the worst case (9th), it requires 19 iterations. It is shown that the refinement steps allow finding the optimal solution in all cases. Regarding the objective function, the proposed strategy slightly overestimates its value by less than 1.1 %. Figure 14 also shows the CPU time breakdown of the ten cases. The average total CPU time is 16 minutes and the most time-consuming step is the adaptive sampling (74-81 %). For the generation of initial sampling data points (1000 points), the CPU time is around 22 minutes (not shown in Fig. 14). Explicit equations obtained with ALAMO are provided in the Supplementary Material for the last run in Fig. 14.

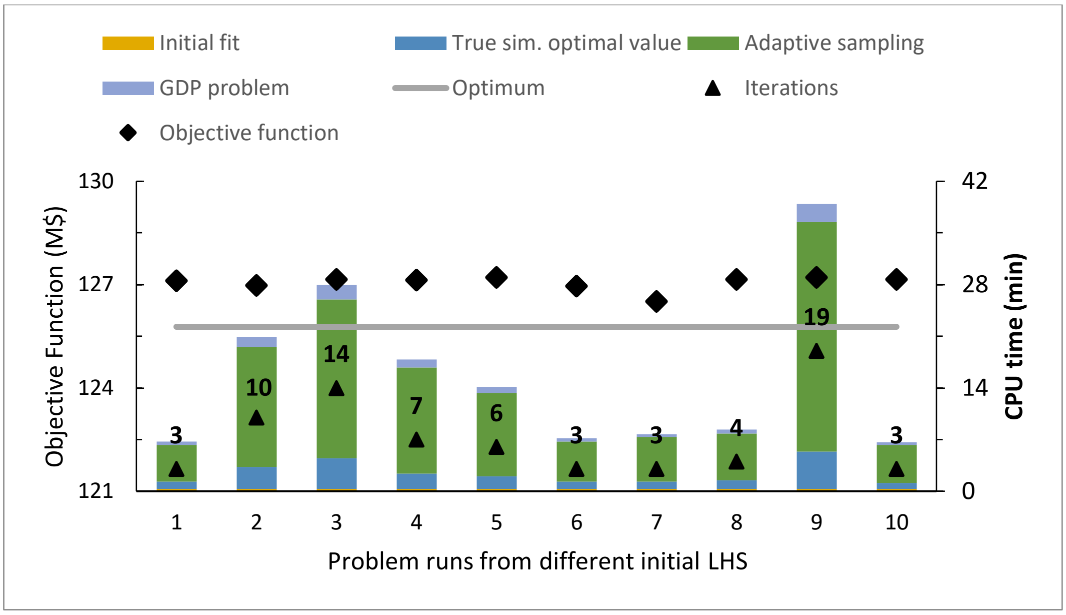


Figure 14: Propylene synthesis problem constrained with hybrid models based on simple algebraic regression functions and GRBF (1000 initial sampling points with LHS)

To assess the impact of the number of data for initial sampling, the problem is solved using 2000 initial points and results are shown in Fig. 15. In this case, three major iterations were required five times, while in the worst case, 10 iterations were necessary. Regarding the objective function value, similar values to those shown in Fig. 14 are obtained. They slightly overestimate the true optimum (in around 1 % in the worst case). The total CPU time distribution is also shown in Fig. 15, with a mean of 11 minutes, with a 27 % reduction with respect to the case of using 1,000 initial data points. However, the CPU time associated with the initial data generation increases 63% (not shown in Fig. 15).

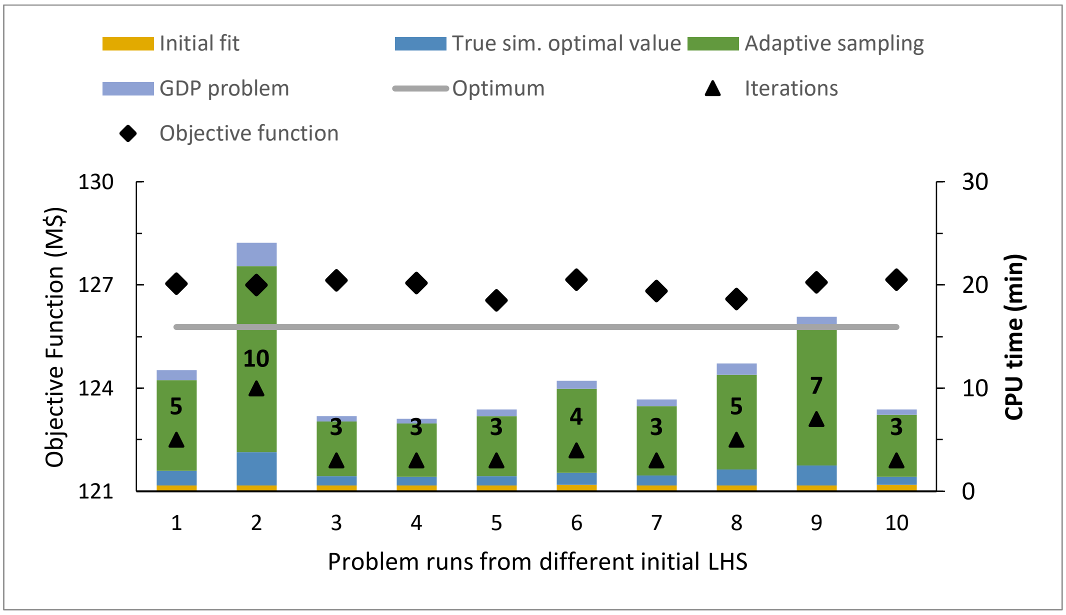


Figure 15: Propylene synthesis problem solved ten times using surrogate model based on both, simple algebraic regression functions and GRBF, considering 2000 initial sampling points

4.4.2 Comparison of optimal design obtained with rigorous and hybrid models

In this section, we report the detailed GDP iterations for the optimal design problem for the propylene production plant, constrained with a hybrid model, as well as a comparison of main variable values with those obtained in the GDP subject to the rigorous model. Table 7 shows iterations for the last GDP problem in Fig. 14 (10th case), which from the ten different LHS designs for initial sampling to generate surrogate models, shows the largest difference in the objective function with the rigorous model in the optimal value (1.1%). The optimal scheme corresponds, as stated before, to the same optimal scheme provided by the GDP subject to the rigorous model. An important reduction in the number of equations is obtained with the use of surrogate models (1,522 vs. 6,547 for the first NLP, 1,379 vs. 6,395 for the second NLP; 2,568 vs. 12,615 for the first MILP, and 3,472 vs. 20,644 for the second Master MILP), and consequently, total CPU time for the GDP solution is 9 s (the solution with the rigorous model required 176 s). These results highlight that one important feature of the proposed procedure is to reduce the GDP problem complexity, while keeping its accuracy.

Table 7: Logic-based Outer Approximation algorithm iterations for optimal design of propylene plant with hybrid model (last GDP in case 10, Fig.14)

Iteration/subproblem	Objective (MM$)	CPU time (s)	Constraints	Continuous vars.	Binary vars.
NLP 1 (CONOPT)	87.36	1.68	1,522	1,423	-
Master MILP 1 (Cplex)	172.00	0.27	2,568	2,740	34
NLP 2 (CONOPT)	127.15	3.32	1,379	1,282	-
Master MILP 2 (Cplex)	147.82	0.42	3,472	3,703	34
NLP 3 (CONOPT)	126.01	3.24	1,379	1,282	-

Furthermore, we compare optimal values obtained for surrogate models input and output variables in both GDPs, the one constrained with the rigorous model and the one with the hybrid model. Table 8 shows a comparison of main variables for the deethylenizer column (DC1), in which the largest relative difference is 6.85% in ethylene component flowrate. As the ethylene feed flowrate is the same in both models, this difference is associated with a higher ethylene conversion estimated in the hybrid model (24 %), with respect to the rigorous model (23 %). A 0.07 % relative difference is obtained in the propylene flowrate. As DC1 is located after DC2, flowrates of C_4^+ components ($f_{DC1,trans\text{-}2\text{-}C_4H_8}^{in}$, $f_{DC1,cis\text{-}2\text{-}C_4H_8}^{in}$, $f_{DC1,1\text{-}C_4H_8}^{in}$, $f_{DC1,C_4H_{10}}^{in}$, $f_{DC1,C_5H_{10}}^{in}$, $f_{DC1,C_6H_{12}}^{in}$) are negligible. Regarding DC1 surrogate model output variables in the optimal design, relative differences of capital cost and reboiler duty are less than 5 % and can be associated with the lower ethylene flowrate fed to DC1 in the hybrid model. DC1 top and bottom temperatures present a good agreement (relative errors less than 0.1 %). Comparison of results for depropylenizer (DC2) and debutanizer (DC3) are provided in the Supplementary Material.

Table 8: Comparison of GDP optimal variable values constrained with rigorous and hybrid models for deethylenizer (DC1), respectively

SM input variables	Original model	Hybrid	Relative difference
$f_{DC1,C_2H_4}^{in}(mol/s)$	677.44	631.05	6.85%
$f_{DC1,C_3H_6}^{in}(mol/s)$	417.04	416.74	0.07%
$Hs_{DC1}(kJ/mol)$	40.29	39.81	1.19%
$P_{DC1}^{top}(bar)$	24.00	24.00	0.00%
SM output variables			
$CC_{DC1}(\$MM)$	4.92	4.82	2.17%
$Q_{DC1}^{reb}(MW)$	4.15	3.95	4.81%
$T_{DC1}^{top}(K)$	250.00	250.16	0.06%
$T_{DC1}^{bottom}(K)$	332.39	332.25	0.04%

4.5 Sensitivity analysis

The effect of ethylene price on the net present value (NPV) is assessed through a sensitivity analysis, as shown in Fig. 16. NPV decreases as ethylene price increases. There is an optimal scheme change from technology A to B (shown in black letters for the true model in Fig. 16). Process B corresponds to the one shown in Fig. 13. Process A uses only ethylene as raw material and includes ethylene dimerization (RDIM), preheater (HXV1) and no butene input stream, so no C4 hydrogenation is selected (C4H). The configuration change from A to B is associated with increasing ethylene cost (higher than 1,000 \$/t). It must be noted that the GDP subject to the rigorous model determines alternative configurations that seem to be suboptimal solutions, maybe due to the high nonlinearities in the process model. Process A^1 includes an additional compressor C3 to DC2 top stream. Process A^2 additionally includes C3 and heat exchanger HXW3. On the other hand, Process B^1 selects a deethylenizer first configuration, which is the only difference with Process B. Figure 16 also shows the number of infeasible NLP subproblems in the L-bOA algorithm for each price scenario. The rigorous model for propylene production via olefin metathesis is large and highly nonlinear, and NLP subproblems do not converge in many cases. In particular, in the first (600 \$/t) and fourth (900 \$/t) price scenarios, there are eight and nine infeasible NLP subproblems, respectively. This analysis is also performed by solving the optimal design problem with the proposed hybrid model generation approach. In this case, we build a hybrid model, and rather than solving a single GDP problem for a given price scenario as shown in the algorithm from Fig. 1, we solve GDP problems for the nine price scenarios, constrained with the same hybrid model. In this way, the NLP subproblem solutions from each GDP price scenario are exploited simultaneously in the algorithm to refine surrogate models and build a new hybrid model for the next major iteration of the proposed algorithm. Regarding NLP subproblem infeasibility, the GDP constrained with the hybrid model did not present any, suggesting that this approach is numerically more robust. Furthermore, the hybrid model approach is computationally more efficient than the rigorous model for this analysis (40.7 min to obtain the optimal solution for all price scenarios, against 60.4 minutes with the rigorous model). Finally, it must be noted that NPV for the optimal design obtained using the rigorous and hybrid models overlap in Fig. 16.

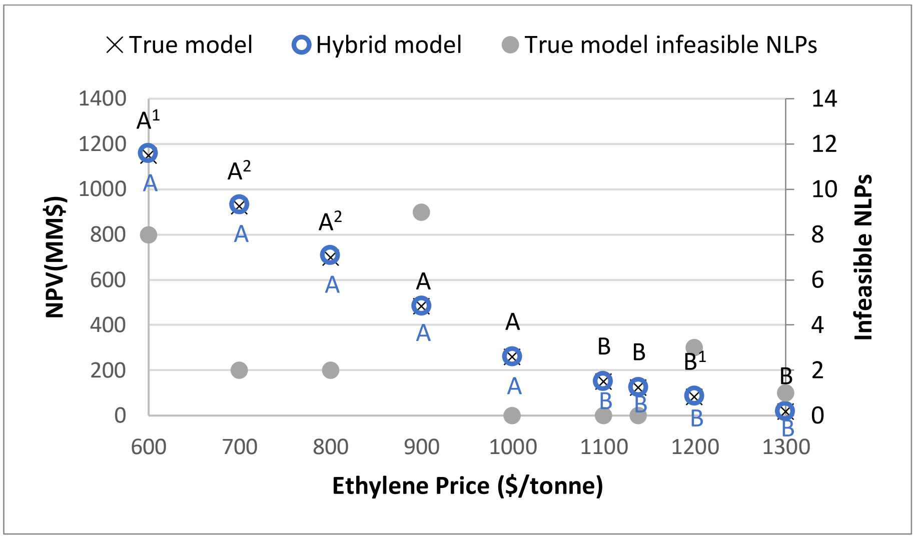


Figure 16: Net present value (NPV) sensitivity to ethylene price for optimal design obtained solving GDP constrained with rigorous and hybrid models. A: Ethylene is the single feedstock. B: Ethylene and butane are feedstock.

5 CONCLUSIONS

In this work, we have proposed a novel iterative procedure for hybrid model generation within a superstructure optimization framework. To build surrogate models, we apply the Latin Hypercube Sampling technique for initial data generation, followed by an initial fit model based on low-complexity algebraic regression functions. For further refinement, Gaussian Radial Basis Functions (GRBFs) are added where the performance of the algebraic regression model is not accurate enough through an adaptive sampling technique (exploration step).

We formulate the optimal design problem for a given superstructure as a Generalized Disjunctive Programming (GDP) problem, and solve the optimization problem with a hybrid model with a custom implementation of the L-bOA algorithm in GAMS. We perform an iterative procedure to continue refining the surrogate model with GRBF (exploitation step) in the domain regions by using the information of the L-bOA solution until convergence criteria are fulfilled. Two case studies were considered with the proposed methodology: a methanol synthesis process and a propylene via olefin metathesis plant design. In each case study, a GDP constrained with a rigorous model is first solved, and then followed by the proposed algorithm for the generation of hybrid models where the GDP constrained with these hybrid models is solved.

In all cases, the same optimal process scheme is determined with both approaches, with relative differences in the objective function lower than 1%. A sensitivity analysis is performed for the net present value of the propylene production plant with respect to raw material price, which supports the conclusion that the proposed methodology provides a more robust approach to address the optimal design of highly nonlinear, large-scale processes. In this analysis, as several scenarios were run, the proposed approach has proven to be computationally more efficient than the use of a rigorous model (with 33% lower CPU time requirement).

6 NOMENCLATURE

6.1 Functions

h^s: Conditional surrogate model function included within a disjunctive term
g^s: Permanent surrogate model function h_i^a: Simple algebraic function i that belongs to a surrogate model function
h^r: Original model function
g^r: Original model constraints

6.2 Variables

CC_u: column capital cost
Q_u^{reb}: reboiler duty
T^{top}: column top temperature
T^{bottom}: column bottom temperature$v$: Surrogate input variables
v_i: Surrogate input variables
FT_u^{in}: Input total flowrate of unit u
FT_u^{out}: Output total flowrate of unit u
$f_{u,\mathrm{j}}^{in}$: Input molar flow of component j of unit u
$f_{u,\mathrm{j}}^{out}$: Output molar flow of component j of unit u
T_u^{in}: Unit u input stream temperature
T_u^{out}: Unit u output stream temperature
P_u^{out}: Unit u output stream pressure
χ_u: Unit u conversion
χ_u^{eq}: Unit u equilibrium conversion
r_u: reactor u hydrogen consumption
$DF_{u,n,j}$: Input flowrates to column u in stage n of component j
$DH_{u,n}$: Input enthalpy flow to column u in stage n
$Dfv_{u,n,j}$: vapor molar flow of component j at stage n in column u
$Dfl_{u,n,j}$: liquid molar flow of component j at stage n in column u

$DV_{u,n}$: vapor total molar flow at stage n in column u
$DL_{u,n}$: liquid total molar flow at stage n in column u
$DR_{u,n}$: liquid-to-vapor ratio at stage n
$DCt_{u,n}$: temperature at stage n
$DCp_{u,n}$: pressure at stage n
$DLp_{u,n,j}$: logarithm of pure component j partial pressure at stage n;
$Dhl_{u,n}$: liquid enthalpy flow from stage n
$Dhv_{u,n}$: vapor enthalpy flow from stage n
$Dhvc_{u,n,j}$: vapor enthalpy of pure component j from stage n
$Dhlc_{u,n,j}$: liquid enthalpy of pure component j from stage n
Q_u^{reb}: reboiler duty in column u
Q_u^{cond}: condenser duty in column u
$CcUt_u$: Unit u cooling utility
$ChUt_u$: Unit u heating utility
CC_u: Unit u capital cost
Hs_u: specific enthalpy of column u input stream
H_u^{in}: total enthalpy of column u input stream
H_u^{top}: enthalpy of column u top stream
H_u^{bottom}: enthalpy of column u bottom stream
$f_{u,j}^{bottom}$: bottom flowrate of component j in column u
T_u^{bottom}: bottom temperature of column u
T_u^{top}: top temperature of column u
P_u^{top}: top pressure of column u
P_u^{bottom}: bottom pressure of column u
$f_{u,j}^{top}$: top flowrate of component j in column u

6.3 Boolean variables

Y_{jk}: Boolean variable that it is true if the k-th term of the j-th disjunction is selected and false otherwise

6.4 Parameters

c_i: surrogate model coefficient i
w_i: surrogate model coefficient i
vs_i: interpolation point i
v_i^*: i-th input variable of the NLP subproblem solution from hybrid GDP
v^L, v^U: lower and upper bounds for surrogate model input variables
N_{input} is the number of surrogate model input variables
ΔH_{rxn}: reaction heat
H_j^0: component enthalpy evaluated at the reference temperature
T^{ref}: reference temperature
$c_{j,i}^{CP}$: polynomial coefficient i to calculate specific heat and vapor enthalpy for component j
$c_{j,i}^{HL}$: polynomial coefficient i to calculate liquid enthalpy for component j
$c_{j,i}^{AN}$: coefficient i of extended Antoine equation for component j
Greek letters
γ: form factor of radial basis functions

6.5 Sets

U: set of units
CD_u: condenser stage in unit u
RB_u: reboiler stage in unit u
N_u^{Feed}: Feed stage in column u
NS_u: separation stages in distillation column u
UV_u: subset of units to which vapor stream enters from unit u
UL_u: subset of units to which liquid stream enters from unit u
Udc: set of columns
J: set of components

6.6 Indices

j: index of components
u: index of units
n: index of column stages
t: index of tasks
r: index of reactions
p: index of phases
m: index of physical change

ACKNOWLEDGMENTS

Support is acknowledged to Consejo Nacional de Investigaciones Científicas y Tecnológicas (Grant no. PIP-2015–11220150100742), Agencia Nacional de Promoción Científica y Tecnológica (Grant no. PICT-2015–3512) and Universidad Nacional del Sur (Grant no. PGI 24/M141). Support is also acknowledged by the Institute for the Design of Advanced Energy Systems (IDAES), U.S. Dept. Energy, Office of Fossil Energy.

7 REFERENCES

Amouzgar, K., Strömberg, N., 2017. Radial basis functions as surrogate models with a priori bias in comparison with a posteriori bias. Struct. Multidiscip. Optim. https://doi.org/10.1007/s00158-016-1569-0

Bajaj, I., Iyer, S.S., Faruque Hasan, M.M., 2018. A trust region-based two phase algorithm for constrained black-box and grey-box optimization with infeasible initial point. Comput. Chem. Eng. 116, 306–321. https://doi.org/10.1016/j.compchemeng.2017.12.011

Bhosekar, A., Ierapetritou, M., 2018. Advances in surrogate based modeling, feasibility analysis, and optimization: A review. Comput. Chem. Eng. 250–267. https://doi.org/10.1016/j.compchemeng.2017.09.017

Biegler, L.T., Grossmann, I.E., Westerberg, A.W., 1997. Systematic Methods of Chemical Process Design, New Jersey. Prentice Hall PTR.

Boukouvala, F., Floudas, C.A., 2017. ARGONAUT: AlgoRithms for Global Optimization of coNstrAined grey-box compUTational problems. Optim. Lett. 11, 895–913. https://doi.org/10.1007/s11590-016-1028-2

Boulamanti, A., Moya, J.A., 2017. Production costs of the chemical industry in the EU and other countries: Ammonia, methanol and light olefins. Renew. Sustain. Energy Rev. 68, 1205–1212. https://doi.org/https://doi.org/10.1016/j.rser.2016.02.021

Caballero, J.A., Grossmann, I.E., 2008. An algorithm for the use of surrogate models in modular flowsheet optimization. AIChE J. 54, 2633–2650. https://doi.org/https://doi.org/10.1002/aic.11579

Chen, Q., Bernal, D.E., Johnson, E.S., Kale, S., Bates, J., Siirola, J.D., Grossmann, I.E., 2020. Pyomo. GDP: an ecosystem for logic based modeling and optimization development.

Online: http://egon.cheme.cmu.edu/Papers/Chen_pyomo_gdp_v6.pdf.

Chen, Q., Grossmann, I., 2019. Modern Modeling Paradigms Using Generalized Disjunctive Programming. Processes 7, 839. https://doi.org/https://doi.org/10.3390/pr7110839

Chen, W., Fu, Z.J., Chen, C.S., 2014. Recent advances in radial basis function collocation methods, in: SpringerBriefs in Applied Sciences and Technology. https://doi.org/10.1007/978-3-642-39572-7

Cozad, A., Sahinidis, N. V, Miller, D.C., 2014. Learning surrogate models for simulation-based optimization. AIChE J. 60, 2211–2227. https://doi.org/https://doi.org/10.1002/aic.14418

Dowling, A.W., Biegler, L.T., 2015. A framework for efficient large scale equation-oriented flowsheet optimization. Comput. Chem. Eng. 72, 3–20.

Fang, H., Rais-Rohani, M., Liu, Z., Horstemeyer, M.F., 2005. A comparative study of metamodeling methods for multiobjective crashworthiness optimization. Comput. Struct. 83, 2121–2136.

Ferris, M.C., Jain, R., Dirkse, S., 2011. Gdxmrw: Interfacing gams and matlab. Online http//www. gams. com/dd/docs/tools/gdxmrw. pdf.

Green, D.W., Perry, R.H., 2007. Perry's Chemical, Perrys' chemical engineers' handbook. https://doi.org/10.1036/0071511245

Henao, C.A., Maravelias, C.T., 2011. Surrogate-based superstructure optimization framework. AIChE J. 57, 1216–1232. https://doi.org/10.1002/aic.12341

Henao, C.A., Maravelias, C.T., 2010. Surrogate-based process synthesis, in: Computer Aided Chemical Engineering. Elsevier, pp. 1129–1134.

Kim, S.H., Boukouvala, F., 2020. Surrogate-Based Optimization for Mixed-Integer Nonlinear Problems. Comput. Chem. Eng. 106847. https://doi.org/https://doi.org/10.1016/j.compchemeng.2020.106847

Kong, L., Sen, S.M., Henao, C.A., Dumesic, J.A., Maravelias, C.T., 2016. A superstructure-based framework for simultaneous process synthesis, heat integration, and utility plant design. Comput. Chem. Eng. 91, 68–84.

McDonald, D.B., Grantham, W.J., Tabor, W.L., Murphy, M.J., 2007. Global and local optimization using radial basis function response surface models. Appl. Math. Model. https://doi.org/10.1016/j.apm.2006.08.008

Mencarelli, L., Chen, Q., Pagot, A., Grossmann, I.E., 2020. A review on superstructure optimization approaches in process system engineering. Comput. Chem. Eng. 106808. https://doi.org/https://doi.org/10.1016/j.compchemeng.2020.106808

Pedrozo, H.A., Reartes, S.B.R., Vecchietti, A.R., Diaz, M.S., Grossmann, I.E., 2021. Optimal Design Of Ethylene And Propylene Coproduction Plants With Generalized Disjunctive Programming And State Equipment Network Models. Comput. Chem. Eng. 107295.

Pedrozo, H.A., Rodriguez Reartes, S.B., Chen, Q., Diaz, M.S., Grossmann, I.E., 2020. Surrogate-model based MILP for the optimal design of ethylene production from shale gas. Comput. Chem. Eng. 141. https://doi.org/10.1016/j.compchemeng.2020.107015

Pedrozo, H A, Rodriguez Reartes, S.B., Diaz, M.S., Vecchietti, A.R., Grossmann, I.E., 2020. Coproduction of Ethylene and Propylene based on Ethane and Propane Feedstocks, in: Computer Aided Chemical Engineering. Elsevier, pp. 907–912. https://doi.org/https://doi.org/10.1016/B978-0-12-823377-1.50152-X

Rios, L.M., Sahinidis, N. V, 2013. Derivative-free optimization: a review of algorithms and comparison of software implementations. J. Glob. Optim. 56, 1247–1293. https://doi.org/https://doi.org/10.1007/s10898-012-9951-y

Sahinidis, N. V, 2019. Mixed-integer nonlinear programming 2018. Optim. Eng. 20, 301–306. https://doi.org/https://doi.org/10.1007/s11081-019-09438-1

Türkay, M., Grossmann, I.E., 1996. Logic-based MINLP algorithms for the optimal synthesis of process networks. Comput. Chem. Eng. 20, 959–978. https://doi.org/https://doi.org/10.1016/0098-1354(95)00219-7

Ulrich, G.D., Vasudevan, P.T., 2006. How to estimate utility costs. Chem. Eng 113, 66–69.

Vecchietti, A., 2011. LogMip [WWW Document]. URL http://www.logmip.ceride.gov.ar/ (accessed 8.1.20).

Vecchietti, A., Grossmann, I.E., 2000. Modeling issues and implementation of language for disjunctive programming. Comput. Chem. Eng. 24, 2143-2155. https://doi.org/10.1016/S0098-1354(00)00582-2

Vecchietti, A., Grossmann, I.E., 1999. LOGMIP: A disjunctive 0-1 non-linear optimizer for process system models, in: Computers and Chemical Engineering. https://doi.org/10.1016/S0098-1354(98)00293-2

Viswanathan, J., Grossmann, I.E., 1993. Optimal feed locations and number of trays for distillation columns with multiple feeds. Ind. Eng. Chem. Res. 32, 2942–2949. https://doi.org/https://doi.org/10.1021/ie00023a069

Wilson, Z.T., Sahinidis, N. V, 2017. The ALAMO approach to machine learning. Comput. Chem. Eng. 106, 785–795. https://doi.org/https://doi.org/10.1016/j.compchemeng.2017.02.010

Yeomans, H., Grossmann, I.E., 1999. Nonlinear disjunctive programming models for the synthesis of heat integrated distillation sequences. Comput. Chem. Eng. 23, 1135–1151. https://doi.org/10.1016/S0098-1354(99)00279-3