

# Surrogate-Based Superstructure Optimization Framework

Carlos A. Henao and Christos T. Maravelias

Dept. of Chemical and Biological Engineering, University of Wisconsin, Madison, WI 53706

DOI 10.1002/aic.12341

Published online July 22, 2010 in Wiley Online Library (wileyonlinelibrary.com).

*In principle, optimization-based “superstructure” methods for process synthesis can be more powerful than sequential-conceptual methods as they account for all complex interactions between design decisions. However, these methods have not been widely adopted because they lead to mixed-integer nonlinear programs that are hard to solve, especially when realistic unit operation models are used. To address this challenge, we develop a superstructure-based strategy where complex unit models are replaced with surrogate models built from data generated via commercial process simulators. In developing this strategy, we study aspects such as the systematic design of process unit surrogate models, the generation of simulation data, the selection of the surrogate’s structure, and the required model fitting. We also present how these models can be reformulated and incorporated into mathematical programming superstructure formulations. Finally, we discuss the application of the proposed strategy to a number of applications.* © 2010 American Institute of Chemical Engineers *AIChE J.*, 57: 1216–1232, 2011

**Keywords:** process synthesis, process optimization, surrogate models

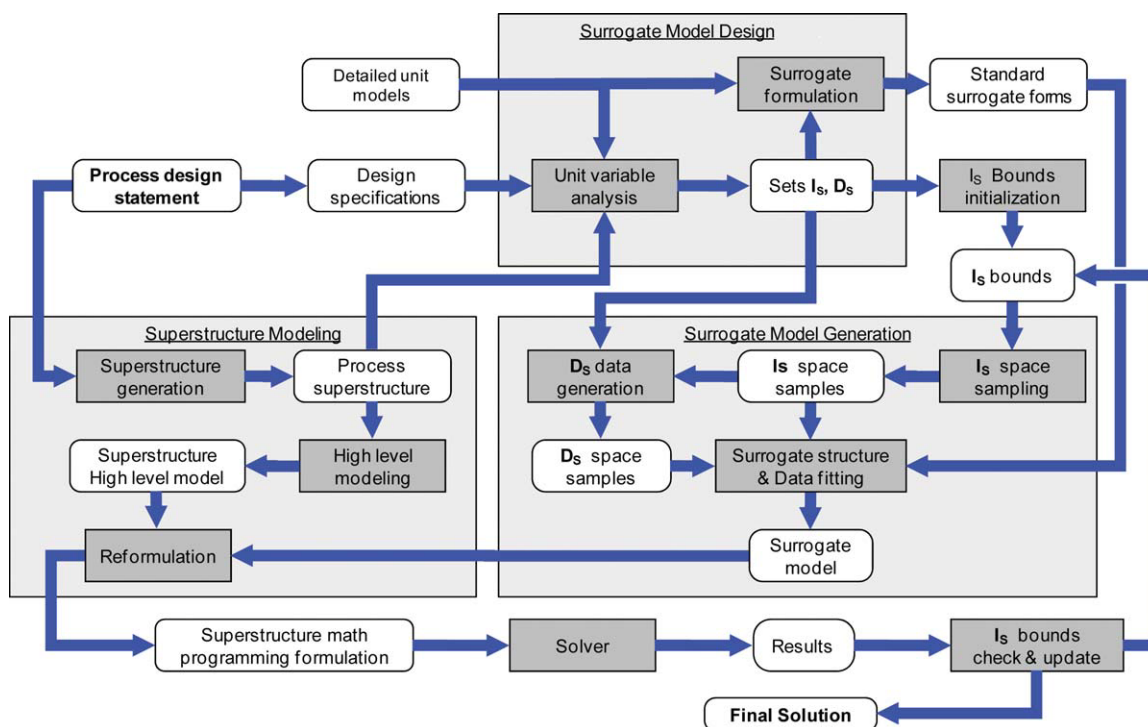
## Introduction

Current chemical processes synthesis methodologies can be classified in two broad categories: the more traditional sequential-conceptual methods and the more systematic superstructure optimization-based methods. Sequential methods<sup>1–4</sup> are based on the existence of a natural hierarchy among the engineering decisions made during the generation of a chemical process flow-sheet. Such an approach is highly popular, because it reduces the complexity of the synthesis problem, leading to a procedure where the main subsystems of the plant are designed sequentially, disregarding the full interaction between decisions made at different stages. Specific conceptual tools have also been developed to support the design of particular plant subsystems (e.g., reactor networks, separation systems, heat recovery networks) by using tailored graphic representations based on thermodynamic insights. This is the case in the design of reaction systems

using the concept of attainable region,<sup>5,6</sup> the design of distillation-based separation systems using residue curves,<sup>7,8</sup> and the design of heat and mass exchanger networks using pinch technology.<sup>9,10</sup>

On the other hand, in superstructure optimization-based methods,<sup>11,12</sup> a network composed by all potentially useful unit operations and all relevant interconnections among them is initially considered. Such superstructure is then used to formulate an optimization model which includes reformulated unit models, interconnection equations, and other constraints, such as thermodynamic property calculation equations. The model typically includes binary selection variables to allow the activation/deactivation of every unit, though other switching methods can be used. The solution of such problem ultimately indicates which of the initially considered units and interconnections are to be kept as well as the values of the optimal operational conditions. A variety of methodologies have been developed for the generation of superstructures in the design of specific plant subsystems, such as reactor networks,<sup>13–15</sup> separation networks,<sup>16–19</sup> and heat exchanger networks.<sup>20–23</sup> Also, alternative methodologies to create and represent general process superstructures have been proposed.<sup>24–29</sup>

Correspondence concerning this article should be addressed to C. T. Maravelias at christos@engr.wisc.edu.



**Figure 1. Overview of the proposed methodology. Sets  $I_s$  and  $D_s$  contain, respectively, surrogate independent and dependent variables.**

[Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]

In theory, the superstructure-based synthesis approach can be very effective because it considers a large number of process alternatives and it involves the simultaneous optimization of the structure and the operational conditions of the process. But this power comes at a price: the mathematical complexity of the resulting optimization model; generally, a large-scale nonconvex mixed-integer nonlinear program (MINLP). In fact, when realistic unit operation models are used (i.e., the kind of models featured in commercial process simulators considering nonideal thermodynamics, kinetics, and transport properties calculations), then the resulting models cannot be solved. Some attempts to reduce the complexity of superstructure optimization models include special modeling/decomposition techniques<sup>30</sup> and the integration of hierarchical decomposition and MINLP frameworks.<sup>31–33</sup> In these approaches, the resulting MINLP models are computationally tractable by balancing the total number of design variables, process alternatives, and the level of detail of sub-system models.

In an attempt to reduce the complexity of engineering design and optimization, several researchers have proposed frameworks, which use surrogate models (also termed meta-models or reduced order models).<sup>34–37</sup> These models are often generated from data obtained via highly accurate although computationally expensive simulation programs, which capture the behavior of particular engineering systems. Interestingly, the use of simulation information to build simpler models for individual pieces of a large system can also be viewed as a multiscale methodology,<sup>38</sup> an approach, which is widely used in modeling-simulation of complex systems. In the case of chemical process synthesis,

commercial process simulators can be used to generate sets of simulation cases for particular process units. Simulation data can then be used in building general purpose multivariable mappings to serve as surrogates, replacing the original complex unit models in generic process optimization problems. Two popular techniques to build such surrogates are artificial neural networks (ANN)<sup>39</sup> and Kriging interpolators.<sup>40</sup> The literature presents several works applying ANNs to modeling, optimization, and control of chemical process systems.<sup>41–44</sup> Some other articles present Kriging-based techniques to support modeling and optimization of chemical process systems,<sup>45,46</sup> as well the solution of special classes of MINLP models in the synthesis and optimization of chemical process.<sup>47,48</sup> However, the use of surrogates for the solution of superstructure optimization problems has not yet been explored in depth.

The objective of this article is to present a comprehensive methodology for the generation of computationally tractable but yet accurate superstructure optimization models for chemical process synthesis problems through the construction and use of unit surrogate models. The presented methodology allows us to make use of realistic unit operation models, including nonideal thermodynamics, while formulating tractable optimization models. An overview of the proposed methodology is shown in Figure 1, where shaded boxes represent modeling, information processing, and optimization tasks; whereas clear rounded boxes represent information and models generated or provided. The methodology starts with the process design statement from which a superstructure can be generated and a set of design specifications can be identified. This information along with knowledge

regarding the variables included in detailed process unit models is used in a systematic variable analysis to identify the optimum set of independent and dependent variables for every unit surrogate model. Given bounds on the independent variables, we define surrogate domains, which are sampled to generate process unit simulation cases. The results of such simulations are then used to fit multivariable mappings obtaining fully functional surrogate models. Using a high-level modeling approach (e.g., generalized disjunctive programming), we generate a superstructure optimization model. This high-level model is then combined with the previously generated surrogates to produce a reformulated MINLP model, which is solved using a commercial solver. The results are analyzed to determine if the surrogate domains have to be updated. If this is the case, then the surrogate model fitting and the solution of the optimization problem are repeated until no further updating is required.

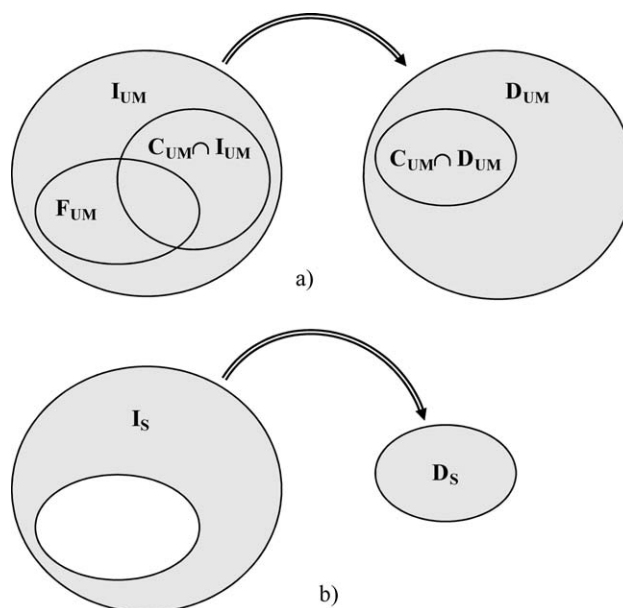
To facilitate the presentation of our strategy, we use the overview in Figure 1 as roadmap. In the next section, we cover the design of surrogates, including a systematic model variable analysis, and its application to the creation of compact formulations for different process units. Then, we present the surrogate model generation, including aspects such as surrogate domain sampling, the generation of simulation data, the surrogate structure selection, and the final data fitting. Next, we discuss superstructure modeling, including alternative generation and representation approaches, as well as the necessary reformulations to incorporate unit surrogate models in superstructure formulations. This is followed by some extensions of the ideas introduced, including the automatic update of surrogate domains, the use of surrogates to describe entire plant subsystems, and the use of surrogates to integrate detailed models developed in different platforms. We close with the presentation of two illustrative example problems.

## Surrogate Model Design

Although several authors have already used surrogate models to address different issues in the solution of optimization problems, no systematic method has been proposed to design compact and accurate surrogates for superstructure-based process synthesis. This section presents a systematic unit model variable analysis to guide the selection of independent and dependent variables and its application to the formulation of standard surrogates for common chemical process units.

### Unit model variable analysis

The replacement of detailed unit models with surrogates is useful if such surrogates include as few of the original variables as possible, while enforcing the same constraints the original model does. The former means that the surrogate is compact, reducing the computational expense in both its generation and later use; whereas the latter ensures that the surrogate is equivalent to the original model in the context of the synthesis problem. To satisfy these two conditions, it is necessary to analyze the original unit models, and the way in which they interact with other units and affect the



**Figure 2. Venn diagrams and representation of the mappings between variable sets.**

a: Original unit model; b: dimensionally reduced surrogate model.

objective function of the synthesis problem. For a particular steady state unit model, we define:

**I<sub>UM</sub>**: Set of independent variables; fixing these variables, transforms the model into a structurally nonsingular equation system.

**D<sub>UM</sub>**: Set of dependent variables (i.e., model variables not in **I<sub>UM</sub>**).

**N<sub>UM</sub>**: Set of *natural* independent variables; these are the variables used in a steady state commercial process simulation model to close its degrees of freedom.

**F<sub>UM</sub>**: Set of variables fixed by the design problem statement (e.g., design specifications).

**C<sub>UM</sub>**: Set of connecting variables; variables in the original unit model that appear in other parts of the optimization problem (e.g., any other unit model, constraint, or the objective function).

In essence, a unit model is a set of equations establishing relationships among the unit variables in such a way that a subset of these variables (**I<sub>UM</sub>**) can be fixed to fully determine the values of the remaining (**D<sub>UM</sub>**). In other words, the model implicitly imposes a mapping from  $Z^I$  (a vector in the space defined by **I<sub>UM</sub>**) to  $Z^D$  (a vector in the space defined by **D<sub>UM</sub>**). For a surrogate model to enforce the same constraints the original model does, it is necessary and sufficient that it characterizes the relationship between the free independent variables **I<sub>UM</sub>/F<sub>UM</sub>** (which in general explains the variability of all dependent variables **D<sub>UM</sub>**) and the connecting variables **C<sub>UM</sub>**. The remaining dependent variables (i.e., **D<sub>UM</sub>/C<sub>UM</sub>**) are internal and required to make the original unit model work, but irrelevant in the context of the process synthesis problem (e.g., temperature profile along a distillation column). Figure 2 shows the Venn diagrams for the variables of the original and surrogate models, along with a graphic representation of the implicit unit model mapping and the dimensionally reduced surrogate model mapping.

For a particular model, the sets  $\mathbf{F}_{\text{UM}}$  and  $\mathbf{C}_{\text{UM}}$  are uniquely determined by the design specifications, the superstructure topology, and the form of the objective function. However, set  $\mathbf{I}_{\text{UM}}$  is not unique because many different subsets of variables can be chosen to close the degrees of freedom of a particular model. Thus, to properly select the variables in  $\mathbf{I}_{\text{UM}}$  we have to account for the structure of the unit model (an equation system). Particularly, a selection of  $\mathbf{I}_{\text{UM}}$  is valid only if fixing those variables results into a unit model that is structurally nonsingular; that is, a square system of equations with no over/under-determined subsystems. We use the following mixed integer programming (MIP) model (M) to systematically select the variables in  $\mathbf{I}_{\text{UM}}$ :

$$\max \sum_{i \in \mathbf{I}} \varphi_i \cdot y_i \quad (1)$$

$$\text{s.t.} \quad \sum_{i: (i,j) \in \mathbf{A}} x_{ij} = 1 \quad \forall j \in \mathbf{J} \quad (2)$$

$$\sum_{j: (i,j) \in \mathbf{A}} x_{ij} = 1 - y_i \quad \forall i \in \mathbf{I} \quad (3)$$

$$x_{ij}, y_i \in \{0, 1\} \quad (4)$$

where  $\mathbf{I}$  and  $\mathbf{J}$  are respectively index sets of variables and equations for a unit model;  $\mathbf{A}$  is the set of edges of the bipartite graph describing the variable-equation incidence structure of the unit model, i.e.  $\mathbf{A} = \{(i,j): \text{variable } z_i \text{ appears in equation } e_j\}$ ;  $x_{ij}$  is a variable-equation matching binary variable;  $y_i$  is a selection binary variable defining  $\mathbf{I}_{\text{UM}}$ , i.e.  $\mathbf{I}_{\text{UM}} = \{z_i: y_i = 1\}$ ; and  $\varphi_i$  is a weight coefficient, which can be adjusted to favor the inclusion in  $\mathbf{I}_{\text{UM}}$  of some variables over others. Constraints (2) and (3) enforce a perfect matching between the model dependent variables,  $z_i \in \mathbf{D}_{\text{UM}}$ , and its equations (i.e. (2)) enforces that every equation is matched with exactly one variable and (3) enforces the matching of every dependent variable with exactly one equation and no matching between independent variable and the equations). This perfect matching ensures the aforementioned structural nonsingularity.<sup>49</sup> Figure 3 illustrates this idea.

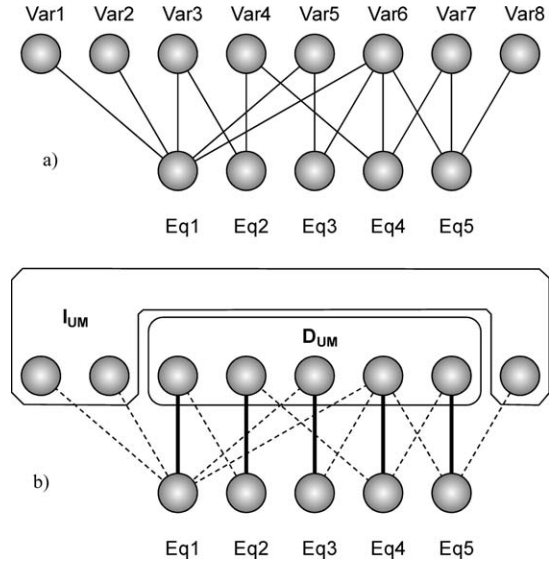
Finally, a couple of important aspects have to be considered when selecting variables in  $\mathbf{I}_{\text{UM}}$ . First, fixed variables in  $\mathbf{F}_{\text{UM}}$  must be included in  $\mathbf{I}_{\text{UM}}$  because, in principle, a fixed variable cannot be dependent. This selection is enforced by making  $y_i = 1 \quad \forall i \in \mathbf{F}_{\text{UM}}$ . If model (M) becomes infeasible in the presence of these additional constraints, then the original specification scheme is inconsistent and has to be reviewed. Second, since every variable in a unit model depends only on the free independent variables ( $\mathbf{I}_{\text{UM}} \setminus \mathbf{F}_{\text{UM}}$ ), and since we are interested in building mappings characterizing only relevant features, the final selection of the surrogate independent (dependent) variable set  $\mathbf{I}_{\text{S}}$  ( $\mathbf{D}_{\text{S}}$ ) is given by (see Figure 2):

$$\mathbf{I}_{\text{S}} = \mathbf{I}_{\text{UM}} \setminus \mathbf{F}_{\text{UM}} \quad (5)$$

$$\mathbf{D}_{\text{S}} = \mathbf{C}_{\text{UM}} \setminus (\mathbf{F}_{\text{UM}} \cup \mathbf{I}_{\text{UM}}) = \mathbf{C}_{\text{UM}} \setminus \mathbf{I}_{\text{UM}} \quad (6)$$

Also, (5) and (6) lead to:

$$|\mathbf{I}_{\text{S}}| = |\mathbf{I}_{\text{UM}}| - |\mathbf{I}_{\text{UM}} \cap \mathbf{F}_{\text{UM}}| = |\mathbf{I}_{\text{UM}}| - |\mathbf{F}_{\text{UM}}| \quad (7)$$



**Figure 3. Equation system representations.**

a) Bipartite representation for a unit model with eight variables and five equations, b) Perfect matching (bold edges) and identification of sets  $\mathbf{I}_{\text{UM}}$ ,  $\mathbf{D}_{\text{UM}}$ .

$$|\mathbf{D}_{\text{S}}| = |\mathbf{C}_{\text{UM}}| - |\mathbf{C}_{\text{UM}} \cap \mathbf{I}_{\text{UM}}| \quad (8)$$

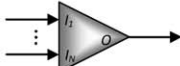
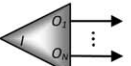
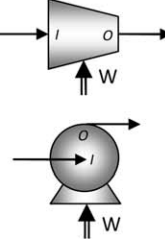
where the second equalities in (6) and (7) consider  $\mathbf{F}_{\text{UM}} \subset \mathbf{I}_{\text{UM}}$ . Note that the set of fixed  $\mathbf{F}_{\text{UM}}$  and connecting  $\mathbf{C}_{\text{UM}}$  variables are well defined once the superstructure has been defined, and  $|\mathbf{I}_{\text{UM}}|$  is also a fixed quantity (e.g., unit model number of degrees of freedom). Thus, (7) and (8) suggest that the number of variables in the surrogate model ( $|\mathbf{I}_{\text{S}}| + |\mathbf{D}_{\text{S}}|$ ) can only be reduced by including in  $\mathbf{I}_{\text{UM}}$  as many of the connecting variables in  $\mathbf{C}_{\text{UM}}$  as possible. Finally, if data to build the surrogate models is obtained from commercial process simulators, it is convenient to include in  $\mathbf{I}_{\text{UM}}$  as many of the natural degrees of freedom  $\mathbf{N}_{\text{UM}}$  as possible. This reduces the use of external specification loops (e.g., adjust functions in HYSYS, process specifications in ASPEN PLUS, etc.) and the associated computational burden. These requirements can be satisfied by selecting large values for  $\varphi_i: z_i \in (\mathbf{C}_{\text{UM}} \cup \mathbf{N}_{\text{UM}})$ .

### Surrogate model formulation

Based on the variable analysis presented in the previous subsection, we developed standard surrogate models for all common process units. The proposed models are given in Tables 1–3, where components, streams, and units are denoted by  $c \in \mathbf{C}$ ,  $s \in \mathbf{S}$ , and  $u \in \mathbf{U}$ , respectively; inlet and outlet streams of unit “ $u$ ” are denoted by  $s \in \mathbf{S}_{\text{in}}^u$  and  $s' \in \mathbf{S}_{\text{out}}^u$ ; variables  $F_{c,s}$ ,  $T_s$ , and  $P_s$ , denote component  $c$  flow rate, temperature and pressure in stream  $s$ ;  $Q_u$  and  $W_u$  denote heating duty and power consumption/generation of unit  $u$ ; whereas  $\Delta P_u$ ,  $\Delta P_{u,s}$ ,  $\xi_{s,u}$ , and  $\varepsilon_u$  denote pressure drop, pressure increase, stream  $s$  split fraction, and isentropic efficiencies of unit  $u$ , respectively. A complete and detailed notation is presented at the end of this article. It is important to note that, in principle, multivariable mappings can be used to replace nonlinear equations for all units where thermodynamic calculations are necessary. With



**Table 1. Surrogate Models for Mixers ( $u \in U^M$ ), Splitters ( $u \in U^S$ ), and Compression-Expression ( $u \in U^{CE}$ ) Units**

<b>Stream Mixers</b>	<b>Stream Splitters</b>	<b>Compressors – Pumps – Turbines</b>
 <p>Mass balances</p> $\sum_{s' \in S_u^O} F_{c,s'} = \sum_{s \in S_u^I} F_{c,s}, \quad \forall c \in C$ <p>Multivariable mapping</p> $T_{s'} = f_{s',u}^T \left( [F_{c,s}, T_s, P_s]_{s \in S_u^I}^{c \in C} \right)$ $P_{s'} = f_{s',u}^P \left( [F_{c,s}, T_s, P_s]_{s \in S_u^I}^{c \in C} \right), \quad \forall s' \in S_u^O$	 <p>Mass balances</p> $F_{c,s'} = F_{c,s} \cdot \xi_{s',u}, \quad \begin{cases} \forall s \in S_u^I \\ \forall s' \in S_u^O \\ \forall c \in C \end{cases}$ $\sum_{s' \in S_u^O} \xi_{s',u} = 1$ <p>Additional expressions</p> $P_{s'} = P_s, \quad \forall s \in S_u^I$ $T_{s'} = T_s, \quad \forall s' \in S_u^O$	 <p>Mass balances</p> $\sum_{s' \in S_u^O} F_{c,s'} = \sum_{s \in S_u^I} F_{c,s}, \quad \forall c \in C$ <p>Additional expressions</p> $P_{s'} = P_s + \Delta P_u, \quad \begin{cases} \forall s \in S_u^I \\ \forall s' \in S_u^O \end{cases}$ <p>Multivariable mapping</p> $T_{s'} = f_{s',u}^T \left( [F_{c,s}, T_s, P_s]_{s \in S_u^I}^{c \in C}, \Delta P_u, \epsilon_u \right), \quad \forall s' \in S_u^O$ $W_u = f_u^W \left( [F_{c,s}, T_s, P_s]_{s \in S_u^I}^{c \in C}, \Delta P_u, \epsilon_u \right)$

this in mind, we developed the general surrogate models given in Tables 1–3. However, if the original models for some of the units are relatively simple, then they can be kept and combined with surrogate models for the remaining units.

To develop these models, we assumed that no variable is fixed (i.e.,  $\mathbf{F}_{UM} = \emptyset$ ) and that the set  $\mathbf{C}_{UM}$  of connecting variables includes the unit input and output streams state variables and the operational variables related to the capital cost and utility usage. Finally, set  $\mathbf{I}_{UM}$  includes only the unit operating variables and the input streams state variables. All these assumptions are necessary to generate general purpose unit surrogate models; however, if necessary, variable analysis with none of these assumptions can be performed to develop more compact models tailored to replace specific units in specific problems.


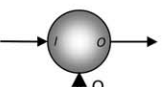
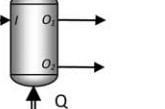
Finally, we have taken advantage of the partial linearity of the original unit models by including some of its equations (mainly the component mass balances and hydraulics) in the surrogate. These equations are used to explicitly calculate a subset of the variables in  $\mathbf{D}_S$  (i.e., outlet component flows and pressure) reducing the set of variables for which the general mapping must be developed. In other words, the proposed surrogates are composed by a subset of linear equations from the original unit model and a reduced multivariable mapping for a subset  $\mathbf{D}_S^* \subset \mathbf{D}_S$ .

To illustrate the generation of standard surrogate models, consider the flash vessel model included in Table 2. The multivariable mapping allows the calculation of the liquid outlet stream component flows  $[F_{c,O_2}]_{c \in C}$  and the unit temperature  $T_u$  from the feed component flows, temperature, and pressure  $[F_{c,S}, T_s, P_s]_{c \in C, s \in S_u^I}$ , as well as the unit pressure drop  $\Delta P_u$  and heating duty  $Q_u$ . The component balances, hydraulics, and thermal equilibrium condition are then used to calculate the vapor outlet stream component flows  $[F_{c,O_1}]_{c \in C}$  and the vapor and liquid outlet stream temperatures and pressures  $[T_s, P_s]_{s \in S_u^O}$ .

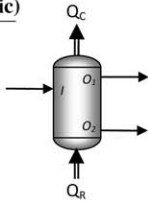
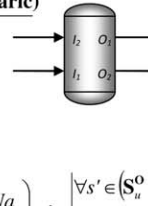
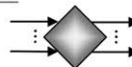
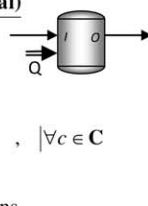
#### Example #1: Tailored unit surrogate model for a homogeneous CSTR

We consider the annual profit maximization of a homogeneous CSTR with fixed feed and pressure drop as presented in Table 4. The reactor operation is constrained by a maximum allowed operational temperature  $T_{\max}$ . The CSTR model includes a general kinetic expression in terms of reactor temperature and molar concentrations, as well as nonideal thermodynamics. The annual profit objective function includes a capital cost term (a function of the reaction volume  $V_R$ ), an operational cost term (a function of the heating duty  $Q_R$ ), and a revenue term (a function of the main product flow leaving the reactor  $F_{c^*,O}$ ).

**Table 2. Surrogate Models for Expansion Valves ( $u \in U^{EV}$ ), Heater-Coolers ( $u \in U^{HC}$ ), and Flash Vessels ( $u \in U^{FV}$ )**

<b>Expansion Valves</b>	<b>Heaters - Coolers</b>	<b>Flash Vessels</b>
 <p>Mass balances</p> $\sum_{s' \in S_u^O} F_{c,s'} = \sum_{s \in S_u^I} F_{c,s}, \quad \forall c \in C$ <p>Additional expressions</p> $P_{s'} = P_s - \Delta P_u, \quad \begin{cases} \forall s \in S_u^I \\ \forall s' \in S_u^O \end{cases}$ <p>Multivariable mapping</p> $T_{s'} = f_{s',u}^T \left( [F_{c,s}, T_s, P_s]_{s \in S_u^I}^{c \in C}, \Delta P_u \right), \quad \forall s' \in S_u^O$	 <p>Mass balances</p> $\sum_{s' \in S_u^O} F_{c,s'} = \sum_{s \in S_u^I} F_{c,s}, \quad \forall c \in C$ <p>Additional expressions</p> $P_{s'} = P_s - \Delta P_u, \quad \forall s \in S_u^I$ $T_{s'} = T_u, \quad \forall s' \in S_u^O$ <p>Multivariable mapping</p> $Q_u = f_u^Q \left( [F_{c,s}, T_s, P_s]_{s \in S_u^I}^{c \in C}, T_u, \Delta P_u \right)$	 <p>Mass balances</p> $\sum_{s' \in S_u^O} F_{c,s'} = \sum_{s \in S_u^I} F_{c,s}, \quad \forall c \in C$ <p>Additional expressions</p> $P_{s'} = P_s - \Delta P_u, \quad \forall s \in S_u^I$ $T_{s'} = T_u, \quad \forall s' \in S_u^O$ <p>Multivariable mapping</p> $F_{c,s'} = f_{c,s',u}^F \left( [F_{c,s}, T_s, P_s]_{s \in S_u^I}^{c \in C}, \Delta P_u, Q_u \right), \quad \begin{cases} \forall s' \in (S_u^O \setminus O_1) \\ \forall c \in C \end{cases}$ $T_u = f_u^T \left( [F_{c,s}, T_s, P_s]_{s \in S_u^I}^{c \in C}, \Delta P_u, Q_u \right)$

**Table 3. Surrogate Models for Distillation Columns ( $u \in U^{DC}$ ), Absorption Columns ( $u \in U^{AC}$ ), and Reactors ( $u \in U^R$ )**

Distillation Columns (Isobaric)	Absorption Columns (Isobaric)	Reactors (Isothermal)
 <p>Mass balances</p> $\sum_{s' \in S_u^0} F_{c,s'} = \sum_{s \in S_u^1} F_{c,s}, \quad  \forall c \in C$ <p>Additional expressions</p> $P_{s'} = P_s - \Delta P_u, \quad \begin{cases} \forall s \in S_u^1 \\ \forall s' \in S_u^0 \end{cases}$ $T_{s'} = T_{cond_u}, \quad  \forall s' \in (S_u^0 \setminus o_2)$ <p>Multivariable mapping</p> $F_{c,s'} = f_{c,s',u}^F \left( \begin{matrix} [F_{c,s'}, T_{s'}, P_{s'}]_{s \in S_u^1}^{c \in C} \\ \Delta P_u, N_{r,u}, N_{s,u}, T_{cond_u}, Br_u \end{matrix} \right), \quad \begin{matrix}  \forall s' \in (S_u^0 \setminus o_1) \\  \forall c \in C \end{matrix}$ $Q_{C,u} = f_u^{Q_C} \left( \begin{matrix} [F_{c,s'}, T_{s'}, P_{s'}]_{s \in S_u^1}^{c \in C} \\ \Delta P_u, N_{r,u}, N_{s,u}, T_{cond_u}, Br_u \end{matrix} \right)$ $Q_{R,u} = f_u^{Q_R} \left( \begin{matrix} [F_{c,s'}, T_{s'}, P_{s'}]_{s \in S_u^1}^{c \in C} \\ \Delta P_u, N_{r,u}, N_{s,u}, T_{cond_u}, Br_u \end{matrix} \right)$ $T_{s'} = f_{s',u}^T \left( \begin{matrix} [F_{c,s'}, T_{s'}, P_{s'}]_{s \in S_u^1}^{c \in C} \\ \Delta P_u, N_{r,u}, N_{s,u}, T_{cond_u}, Br_u \end{matrix} \right), \quad  \forall s' \in (S_u^0 \setminus o_1)$	 <p>Mass balances</p> $\sum_{s' \in S_u^0} F_{c,s'} = \sum_{s \in S_u^1} F_{c,s}, \quad  \forall c \in C$ <p>Multivariable mapping</p> $F_{c,s'} = f_{c,s',u}^F \left( \begin{matrix} [F_{c,s'}, T_{s'}, P_{s'}]_{s \in S_u^1}^{c \in C} \\ \Delta P_u, Na_u \end{matrix} \right), \quad \begin{matrix}  \forall s' \in (S_u^0 \setminus o_1) \\  \forall c \in C \end{matrix}$ $T_{s'} = f_{s',u}^T \left( \begin{matrix} [F_{c,s'}, T_{s'}, P_{s'}]_{s \in S_u^1}^{c \in C} \\ \Delta P_u, Na_u \end{matrix} \right), \quad  \forall s' \in S_u^0$ $P_{s'} = f_{s',u}^P \left( \begin{matrix} [F_{c,s'}, T_{s'}, P_{s'}]_{s \in S_u^1}^{c \in C} \\ \Delta P_u, Na_u \end{matrix} \right), \quad  \forall s' \in S_u^0$ <p>Agregators - Disagregators</p> $\sum_{s' \in S_u^0} F_{c,s'} = \sum_{s \in S_u^1} F_{c,s}, \quad  \forall c \in C$ $\sum_{s' \in S_u^0} T_{s'} = \sum_{s \in S_u^1} T_s, \quad  \forall c \in C$ $\sum_{s' \in S_u^0} P_{s'} = \sum_{s \in S_u^1} P_s$ 	 <p>Mass balances</p> $\sum_{s' \in S_u^0} F_{c,s'} = \sum_{s \in S_u^1} F_{c,s} + \sum_{r \in R} v_{r,c} \cdot X_{r,u}, \quad  \forall c \in C$ <p>Additional expressions</p> $P_{s'} = P_s - \Delta P_u, \quad  \forall s \in S_u^1$ $T_{s'} = T_u, \quad  \forall s' \in S_u^0$ <p>Multivariable mapping</p> $Q_u = f_u^Q \left( \begin{matrix} [F_{c,s'}, T_{s'}, P_{s'}]_{s \in S_u^1}^{c \in C} \\ \Delta P_u, T_u, L_u, D_u \end{matrix} \right)$ $X_{r,u} = f_{r,u}^X \left( \begin{matrix} [F_{c,s'}, T_{s'}, P_{s'}]_{s \in S_u^1}^{c \in C} \\ \Delta P_u, T_u, L_u, D_u \end{matrix} \right), \quad  \forall r \in R$

Our objective is to build a surrogate to replace the detailed CSTR model. The variables of the unit model can be classified as follows:

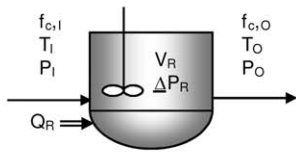
- The problem statement indicates feed conditions and reactor pressure drop are fixed, hence,  $F_{UM} = \{T_I, P_I, [F_{c,I}]_{c \in C}, \Delta P_R\}$ .

- The CSTR model is connected to the objective function through the main product flow, the reactor volume, and heat duty. Additionally, the model is also connected to another constrain trough the reactor temperature; hence,  $C_{UM} = \{T_O, F_{c*,O}, V_R, Q_R\}$ .

- Commonly, in commercial process simulators a CSTR specification set includes the feed stream state variables, unit pressure drop, reaction volume, and operating temperature; hence,  $N_{UM} = \{T_I, P_I, [F_{c,I}]_{c \in C}, \Delta P_R, V_R, T_O\}$ .

As the CSTR model is relatively simple and for this case  $F_{UM} \subset N_{UM}$ , we can determine  $I_{UM}$  without solving problem (M). In fact, by selecting  $I_{UM} = N_{UM}$ , we satisfy all the desirable characteristics of  $I_{UM}$ . Using Eqs. (5) and (6), we obtain  $I_S = \{V_R, T_O\}$  and  $D_S = \{Q_R, F_{c*,O}\}$ . This is a tailored surrogate design, which turns out to be significantly more compact than the standard form presented in Table 3.

**Table 4. Detailed Homogeneous CSTR Simulation and Optimization Models**

CSTROptimization problem	CSTRdetailed model ( $f_{CSTR}(\dots) = 0$ )	
$\max f_{Obj}(F_{c*,O}, V_R, Q_R)$ $V_R, T_O$ $s.t. \begin{cases} f_{CSTR}(\dots) = 0 \\ T_O \leq T_{max} \end{cases}$	<p>Mass balances</p> $F_{c,I} + V_R \cdot \sum_{r \in R} (v_{r,c} \cdot r_r) = F_{c,O}, \quad  \forall c \in C$ <p>Energy balance</p> $h_I \cdot \left( \sum_{c \in C} F_{c,I} \right) + Q_R = h_O \cdot \left( \sum_{c \in C} F_{c,O} \right)$ <p>Additional expressions</p> $P_I - \Delta P_R = P_O$ $r_r = f^r_r(T_O, [C_{c,O}]_{c \in C}), \quad  \forall r \in R$ $x_{c,I} \cdot \left( \sum_{c \in C} F_{c,I} \right) = F_{c,I}, \quad x_{c,O} \cdot \left( \sum_{c \in C} F_{c,O} \right) = F_{c,O}, \quad  \forall c \in C$ $C_{c,O} = x_{c,O} / v_O$	
<p>Partial notation</p> <p><b>C, R</b> : Components and reactions</p> <p><math>F_{c,I}, F_{c,O}</math> : Component flows</p> <p><math>v_{r,c}</math> : Stoichiometry</p> <p><math>r_r</math> : Reaction rates</p> <p><math>h_I, h_O</math> : Molar enthalpies</p> <p><math>v_O</math> : Molar volume</p> <p><math>C_c</math> : Molar concentrations</p>	<p>Thermodynamics</p> $h_I = f^{h_I}(T_I, P_I, [x_{c,I}]_{c \in C})$ $h_O = f^{h_O}(T_O, P_O, [x_{c,O}]_{c \in C})$ $v_O = f^{v_O}(T_O, P_O, [x_{c,O}]_{c \in C})$	

**Table 5. Applicability Region for a Distillation Column Surrogate Model**

Variable	Lower bound	Upper bound
$F_{O_2,I}$ (kmol/h)	0.01	0.1
$F_{N_2,I}$ (kmol/h)	0.01	0.1
$F_{CO_2,I}$ (kmol/h)	5000	10000
$F_{H_2O,I}$ (kmol/h)	50000	100000
$F_{MEA,I}$ (kmol/h)	5000	15000
$T_1$ (K)	300	360
$P_1$ (kPa)	150	200
$\Delta P$ (kPa)	0	0
Nr	1	1
Ns	3	3
$T_{cond}$ (K)	313.15	313.15
Br	0.01	3

## Surrogate Model Generation

In this section, we present several aspects regarding the automatic generation of surrogate models, including the sampling of the independent variable space, the generation of points in the dependent variable space, and the final surrogate-data fitting. All the aforementioned components have been implemented in ASPEN-PLUS and MATLAB and integrated as shown in Figure 4.

### Independent variable sampling

Once the sets of independent ( $I_S$ ) and dependent ( $D_S$ ) variables have been determined for a particular unit, the  $I_S$  space (i.e., the Cartesian product of the domains of variables in  $I_S$ ) can be sampled. Here, the primary objective is the generation of a high amount of information from a limited number of sampling points. In other words, it is important to maximize the sample information content by a proper selection of the sampling points. A way to accomplish this is by selecting points with very different values for every variable, leading to a set of simulation cases where every variable in  $I_S$  has a maximum number of well distributed sampling values. Under this perspective, both grid sampling and random sampling are not appropriate because the former leads to subsets of points, which share values for one or several variables, and the latter can lead to a non homogeneous coverage of the sampled space. To avoid these difficulties, special experimental design techniques, such as Latin Hypercube, Hammersley sequences, and D-optimal designs, have to be used. Among this procedures, the classic Latin Hypercube and its modifications are very popular in general applications,<sup>50</sup> whereas D-optimal designs are advantageous as they account for the type of fitting model during the generation of sampling points.<sup>51</sup> In this work, we developed a MATLAB sampling subroutine that uses a modified Latin Hypercube design to define the sampling matrix of the  $I_S$  space from given bounds on the variables in  $I_S$  (initially provided by the user and later updated by another subroutine).

### Dependent variable data generation

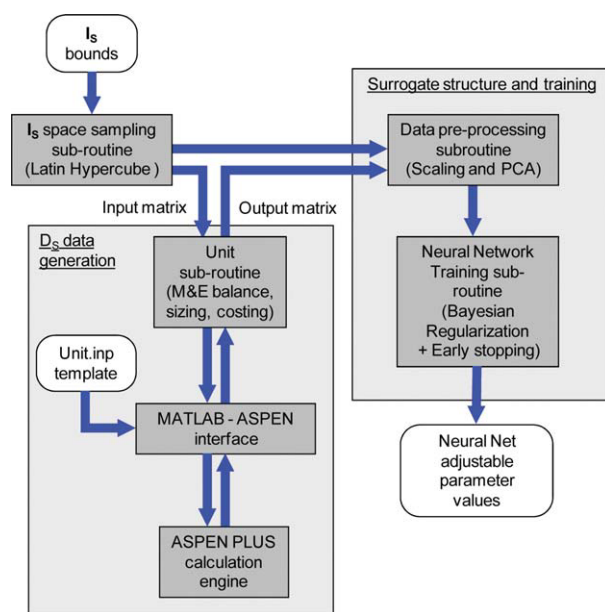
The sampling matrix of the  $I_S$  space is used to specify different simulation cases, which provide the values of the dependent variables  $D_S$ . As part of the implementation for this work, a series of MATLAB unit subroutines, a tailored MATLAB-ASPEN interfacing subroutine, and a series of standard ASPEN simulation models were developed. The

way all these components interact is illustrated in Figure 4. The MATLAB unit subroutines use material and energy (M&E) balance information generated by ASPEN Plus to perform sizing-costing calculations. The MATLAB-ASPEN interface manages the information flow between the MATLAB code and the ASPEN Plus calculation engine. Finally, each of the mentioned standard ASPEN Plus simulation models features a distinct unit operation. All ASPEN simulation input files (i.e., UnitName.inp) have a very similar structure that includes a “sensitivity block paragraph” defining the sets of dependent and independent variables.<sup>52</sup>

To generate the sample values for the dependent variables in  $D_S$  of a particular unit, the corresponding unit subroutine calls the MATLAB-ASPEN interface. This interface calls the proper standard ASPEN simulation input file (e.g., flash.inp, reactor.inp, etc.), rewrites the sensitivity block paragraph to specify simulation cases using the information in the  $I_S$  sample matrix, calls the ASPEN Plus calculation engine on the modified ASPEN input file, and retrieves the material and energy balance values generated by ASPEN Plus in the corresponding output file (e.g., flash.out, reactor.out, etc.). This simulation information is used by the unit subroutine to perform automatic sizing and costing calculations based on standard algorithms.<sup>53</sup> The M&E balances and cost information is presented in a  $D_S$  sample matrix.

### Surrogate structure and data fitting

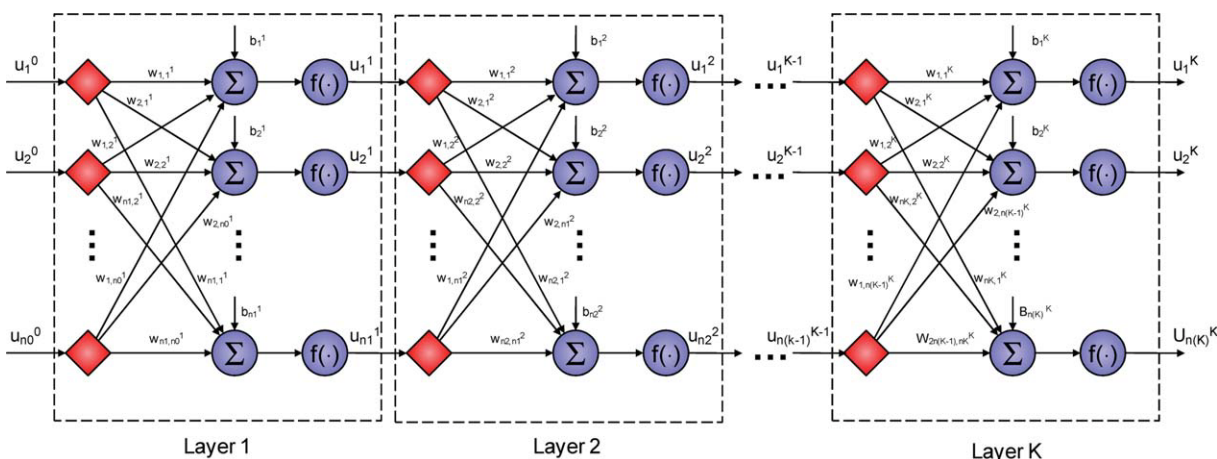
The variable analysis presented above is general and applicable to any kind of detailed process unit and surrogate model; however, the generation of the multivariable mapping depends on the type of surrogate model. Although any such procedure involves the solution of an optimization problem, the approach and the corresponding algorithms are rather diverse. In this work, we used ANN because of their



**Figure 4. Automatic generation of simulation data and ANN training.**

[Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]





**Figure 5. Graphic representation of a multilayer perceptron MLP.**

[Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]

excellent fitting characteristics and low complexity, and because they can be easily reformulated to be incorporated in a superstructure mathematical programming model. Nevertheless, it is important to note that the proposed methodology can be used to develop surrogate models of any type or approximations of parameters of well known short-cut models.

One of the most common ANN configurations is the so-called multilayer perceptron (MLP).<sup>39</sup> A MLP is a highly parallel computational structure composed by a series of “layers”, connected as presented in Figure 5. Mathematically, every layer  $k$  in a MLP acts as a transformation between an input vector  $\mathbf{u}^{k-1}$  and an output vector  $\mathbf{u}^k$  according to:

$$\mathbf{u}^k = f(\mathbf{W}^k \cdot \mathbf{u}^{k-1} + \mathbf{b}^k), \quad k \in \mathbf{K} = \{1, \dots, K\} \quad (9)$$

The set of all relations (9) defines the mapping between the network input  $\mathbf{u}^0$  and its output  $\mathbf{u}^K$ . Parameters  $\mathbf{b}^k$  and  $\mathbf{W}^k$  of layer  $k$  (composed by  $n_k$  neurons) are known as the layer bias vector ( $n_k$ -dimensional) and weight matrix ( $n_k \times n_{k-1}$  dimensional), respectively. The function  $f(\cdot)$  [generally  $\tanh(\cdot)$ ] is known as activation function, and acts on every component of its argument vector to generate the output  $\mathbf{u}^k$ . The last layer ( $k = K$ ) is called the output layer, whereas the rest of them are called hidden layers.

The fitting characteristics of ANNs are described by the following theoretical result<sup>39</sup>: an arbitrary multivariable function with a finite number of discontinuities can be approximated with an arbitrary degree of accuracy using a MLP having only two layers; a hidden layer with a sigmoid activation function (e.g.,  $\tanh(\cdot)$ ), and an output layer with a linear activation function. The exclusive use of this kind of MLPs reduces the identification of the optimal network architecture to the determination of the proper number of neurons in the hidden layer. This is the case since the number of neurons in the outlet linear layer is always equal to the number of mapping dependent variables (i.e.,  $|\mathbf{D}_S^*|$ ).

ANN fitting is referred to as “training.” There are different training algorithms and data preprocessing options that enhance the ANN-fitting performance. In this case, a training algorithm with Bayesian regularization and early stopping was used.<sup>54</sup> In general, the training involves the minimiza-

tion of the sum of squared model deviations with respect to a subset of the available data points called the “training set.” Training with regularization takes as objective function the weighted sum of two terms: the traditional sum of squared deviations and the mean sum of squared network parameters  $w_{ij}^k$  and  $b_j^k$ . The weights of these terms are denoted by  $\gamma$  and  $(1 - \gamma)$ , respectively. This alternative objective function guides the optimization algorithm towards solutions with small network parameters, which are characteristic of smooth fitting surfaces. Bayesian regularization is a statistical framework that allows the automatic determination of the so-called performance ratio  $\gamma$ .

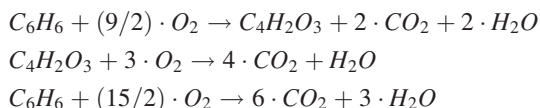
Early stopping is a training termination criterion, and involves the monitoring of the model deviations with respect to a second subset of the available data points known as the “validation set.” Both regularization and early stopping reduce over-fitting, a condition in which the trained network features very small deviations with respect to the training set but big deviations with respect to the validation set, indicating a poor network capacity to reproduce data outside of the training set (i.e., poor interpolation—extrapolation capabilities). This is particularly important when dealing with noisy data and small data sets.

Other important aspects are the initialization algorithm for weights and biases and the way the data is preprocessed. Here, we used the Nguyen-Widrow initialization procedure.<sup>55</sup> The network parameter initialization algorithm distributes the active region of each neuron in a layer evenly through the layer input space, hence increasing the fraction of neurons actively used during the training, the training speed, and the prediction capabilities of the network for a fixed number of neurons.<sup>55</sup> Data scaling and principal component analysis were used to enhance the information content in the data.<sup>54</sup> The data preprocessing and ANN training is performed by the same MATLAB subroutine.

#### **Example #2: Optimum design and operation of a homogeneous CSTR**

We consider the production of maleic anhydride (MA) from Benzene (B) in a CSTR fed with 966 kmol/h of air and 34 kmol/h of benzene at 300 K, 1013 kPa according to the following vapor phase catalytic reactions:





Assuming no pressure drop, the goal is to find the operating temperature and reactor volume maximizing the annualized profit. Reaction kinetics for this reactive system is reported in the literature.<sup>56</sup> Annualized profit includes revenue from MA at current price (www.icis.com), capital cost,<sup>57</sup> and utility costs.

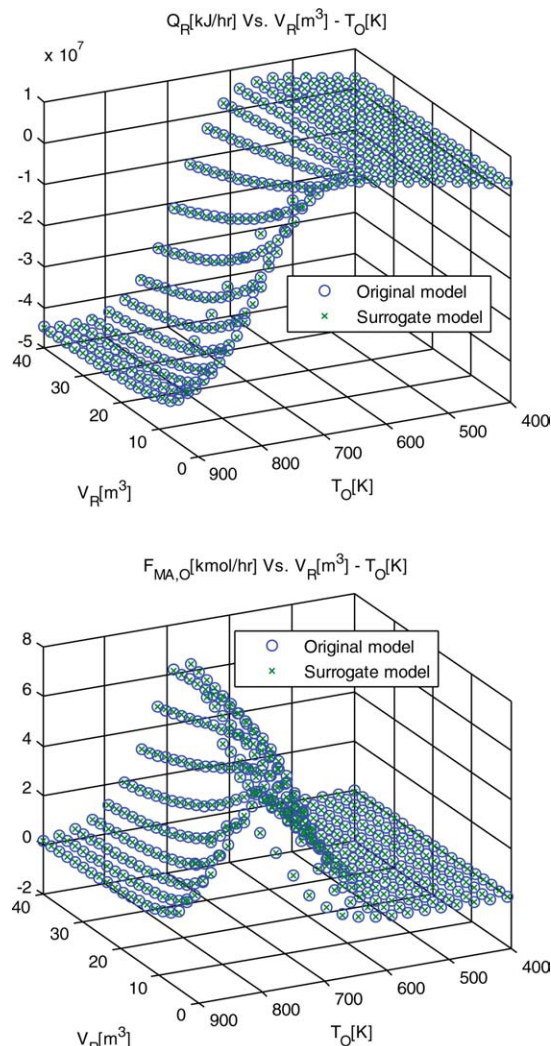
The underlying optimization problem has the same structure as the one presented in Example 1. Following the proposed methodology, we use a surrogate to replace the original CSTR model. Recalling our analysis in Example 1, the detailed CSTR model can be replaced by a mapping from  $\mathbf{I}_S = \{V_R, T_O\}$  (i.e., reactor volume and outlet temperature) into  $\mathbf{D}_S = \{Q_R, F_{MA,O}\}$  (i.e., reactor heating duty and MA outlet flow). We developed an ANN with a linear output layer, two hidden layers with five neurons each, using  $\tanh(\cdot)$  as the activation function. At the sampled points, the maximum surrogate deviation observed in the values of the dependent variables was less than 0.4% of the mapping range. Figure 6 presents the fitting capabilities of the generated surrogate. The optimization problem was solved using CONOPT in GAMS v22.5 to obtain an optimal reactor temperature of 670 K and an optimal reactor volume of 40 m<sup>3</sup>. All calculations were performed in a PC with an Intel(R) Core(TM)i7 CPU 920 @ 2.67 GHz. The time required for the generation of the surrogate was 30.4 s and the time required for the solution of the simplified optimization model was 0.13 s

## Superstructure Generation and Modeling

This section discusses the generation of a process superstructure and the formulation of the corresponding optimization model. It includes a brief comment on the generation of superstructures, followed by a presentation of two popular superstructure mathematical modeling techniques, and the reformulation required to incorporate ANN surrogate models within a superstructure formulation.

### Superstructure generation

One of the critical aspects affecting the quality of a process synthesis solution obtained by a superstructure-based approach is the extent of the design space covered by the original superstructure. Clearly, the optimal solution cannot be found if the optimal structure is not embedded within the original superstructure. Traditionally, superstructure formulation has been tackled using ad-hoc approaches that rely on empirical knowledge and insights. Although several works focus on the design of specific plant subsystems,<sup>13–29</sup> only a hand full of them address the problem of superstructure generation in a rigorous manner. One of these approaches<sup>24–27</sup> uses combinatorial techniques and graph theory to formulate an axiom system, which defines the properties any feasible process structure should satisfy. This set of axioms is complemented by a set of theorems allowing the explicit manipulation of process structures. This



**Figure 6. Original and surrogate model predicted values for  $Q_R$  and  $F_{MA,O}$ .**

[Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]

forms the theoretical basis of the maximal structure generation (MSG).<sup>25</sup>

### Superstructure modeling approach

All approaches to superstructure formulation have something in common: the capacity to deal with the discrete-continuous nature of the synthesis problem, which requires the identification of both the optimal structure (discrete decisions) and operational conditions (continuous decisions).

One powerful approach involves the use of Mathematical Programs with Equilibrium Constraints (MPEC) or Mathematical Programs with Complementarity Constraints (MPCC), where complementarity constraints are used to model disjunctions without using binary variables. This approach is advantageous because it leads to optimization problems, which can be handled by standard NLP solvers,<sup>58</sup> though the effective use of MPECs-MPCCs requires special care in their formulation and solution strategies due to their inherent nonconvexity and lack of NLP regularity properties.<sup>58–60</sup>

In this article, we represent process superstructures as “State Task Networks-One Task One Equipment” (STN-OTOE),<sup>12</sup> and formulate the corresponding optimization problem as a generalized disjunctive program (GDP).<sup>61</sup> The general form of the GDP formulation we consider includes Eqs. 10–13:

$$\max \sum_{s \in \mathbf{S}^{\text{PD}}} \pi_s \cdot \left( \sum_{c \in \mathbf{C}} F_{c,s} \right) - \sum_{s' \in \mathbf{S}^{\text{RM}}} \pi_{s'} \cdot \left( \sum_{c \in \mathbf{C}} F_{c,s'} \right) - \sum_{u \in \mathbf{U}} \left( \left( \frac{A}{P} \right) \cdot CC_u + COP_u \right) \quad (10)$$

$$s.t. \quad \left( \begin{array}{l} h_u \left( [F_{c,s}, T_s, P_s]_{\substack{c \in \mathbf{C} \\ s \in \mathbf{S}_u}}, \psi_u \right) = 0 \\ g_u \left( [F_{c,s}, T_s, P_s]_{\substack{c \in \mathbf{C} \\ s \in \mathbf{S}_u}}, \psi_u \right) \leq 0 \\ CC_u = f_u^{\text{CC}}([T_s, P_s]_{s \in \mathbf{S}_u}, \psi_u), \quad COP_u = f_u^{\text{COP}}(\psi_u) \end{array} \right) \quad \forall u \in \mathbf{U}^{\text{P}} \quad (11)$$

$$\left[ \begin{array}{l} h_u \left( [F_{c,s}, T_s, P_s]_{\substack{c \in \mathbf{C} \\ s \in \mathbf{S}_u}}, \psi_u \right) = 0 \\ g_u \left( [F_{c,s}, T_s, P_s]_{\substack{c \in \mathbf{C} \\ s \in \mathbf{S}_u}}, \psi_u \right) \leq 0 \\ CC_u = f_u^{\text{CC}}([T_s, P_s]_{s \in \mathbf{S}_u}, \psi_u), \quad COP_u = f_u^{\text{COP}}(\psi_u) \end{array} \right] \vee \left[ \begin{array}{l} F_{c,s} = 0 \\ T_s = 0 \\ P_s = 0 \\ \psi_u = 0 \\ CC_u = 0, \quad COP_u = 0 \end{array} \right] \quad \forall u \in \mathbf{U}^{\text{C}} \quad (12)$$

$$\Omega(Y) = \text{True}, Y_u \in \{\text{True}, \text{False}\} \quad (13)$$

The objective function (10) used here is an approximation of the process annual profit, accounting for revenue and raw material costs using molar prices  $\pi_s$  (for product streams  $s \in \mathbf{S}^{\text{PD}}$  and raw material streams  $s' \in \mathbf{S}^{\text{RM}}$ ), as well as the annual cost of the plant, including the annualized capital cost  $(A/P)CC_u$  and operating cost  $COP_u$  of every unit  $u$  in the process. Constraint (11) includes the models of permanent units  $u \in \mathbf{U}^{\text{P}}$  (i.e., units we know will be included in the final solution), whereas constraint (12) includes the models of conditional units  $u \in \mathbf{U}^{\text{C}}$  (i.e., units whose presence within the final solution is to be decided) in the form of disjunctive constraints, where Boolean variables  $Y_u$  are used for the activation/deactivation of entire unit models. In previously proposed methods, unit models in Eqs. 11 and 12 include material and energy balances, kinetics, thermodynamic property expressions, sizing-costing equations, and inequality constraints in the form of relationships among the component flows, temperature, and pressure of the streams  $s \in \mathbf{S}_u$  entering and leaving the particular process unit  $u$ , and its operational variables  $\psi_u$ . Finally, a group of logic constraints (13) in terms of the mentioned Boolean variables  $Y_u$  accounts for superstructure connectivity implications.<sup>62</sup>

Although GDP models can be handled directly by the logic-based solver LogMIP,<sup>63</sup> in the next subsection, we discuss a reformulation of ANN surrogate models that allows us to develop a computationally tractable MINLP reformulation that can be solved using traditional solvers.<sup>64,65</sup>

### Superstructure and surrogate model reformulation

Linear constraints included in disjunctions can be readily reformulated into mixed-integer constraints using well known methods. We propose the use of the convex-hull reformulation to handle the linear equations of our surrogate models<sup>66</sup> (see Tables 1–3). The use of such strategy for non-

linear constraints however results into mixed-integer nonlinear constraints that can lead to computational difficulties.<sup>67,64,65</sup> To address this challenge, we reformulate the nonlinear equations in our surrogate models as follows,

$$\left( \begin{array}{l} \mathbf{u}^k = f(\mathbf{v}^k) \\ \mathbf{v}^k = \mathbf{W}^k \cdot \mathbf{u}^{k-1} + \mathbf{b}^k \end{array} \right)_{k=1, \dots, K} \rightarrow \left( \begin{array}{l} \mathbf{u}^k = f(\mathbf{v}^k) \\ \mathbf{v}^k = \mathbf{W}^k \cdot \mathbf{u}^{k-1} + \mathbf{b}^k y \\ \mathbf{U}_L^{k-1} y \leq \mathbf{u}^{k-1} \leq \mathbf{U}_U^{k-1} y \\ y \in \{0, 1\}, \quad k = 1, \dots, K \end{array} \right) \quad (14)$$

where we use a MLP given by (9); the activation function is  $\tanh(\cdot)$  for the hidden layers and linear for the output layer;  $\mathbf{U}_L^k$  and  $\mathbf{U}_U^k$  are lower and upper bounds on the components of  $\mathbf{u}^k$ ; and  $y$  is a binary selection variable, that is, the binary that corresponds to the Boolean variable  $Y_u$  in Eq. 12.

The above reformulation has two major advantages: (a) unlike standard methods, no variable transformation that results in numerical difficulties is necessary, and (b) the nonlinear equation  $\mathbf{u}^k = \tanh(\mathbf{v}^k)$  does not involve binary variable  $y$ . The development of this reformulation exploits the fact that  $\tanh(0) = 0$ . Thus, when  $y = 1$  the original surrogate relation is enforced; and, when  $y = 0$ , the surrogate model is deactivated, leading to  $\mathbf{u}^0 = 0$ ,  $\mathbf{u}^K = 0$ . The effectiveness of this reformulation is another reason why we chose to use ANN surrogate models. Finally, note that the proposed strategy leads to the replacement of the multiple types of highly nonlinear equations in the original unit models with only one type of nonlinear constraints, namely, the sigmoid  $\tanh(\cdot)$ . Hence, future work will focus on the development of solution methods that exploit this special structure. One alternative includes the use of convexification

techniques similar to the ones presented by Westerlund et al.<sup>68,69</sup> in search for global solutions.

Our surrogate-based MINLP model includes the surrogate models of the permanent units, the reformulation of the surrogate models we just introduced for conditional units, and the algebraic equivalents of the logic constraints in (13).

## Extensions

### Surrogate domain update

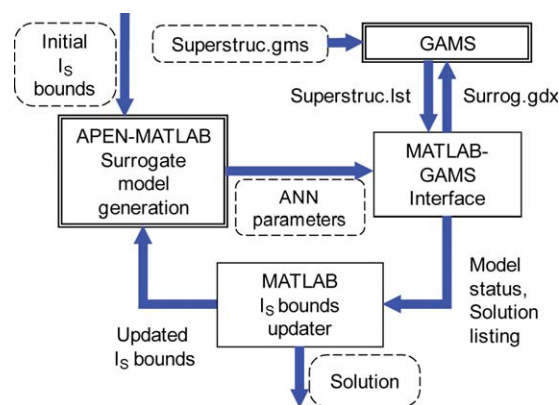
Once the surrogates have been generated, their validity is limited to the sampling domain of their  $I_S$  space, henceforth denoted as surrogate domains. In fact, the bounds of such domains are included in our formulation to ensure that surrogate models are used correctly. However, this can lead to infeasible programs when, for example, two units are interconnected and the surrogate range of the upstream unit for any of the variables in the connecting stream does not overlap the surrogate domain of the downstream unit for the same variable.

To address this challenge, we update the surrogate domains every time an infeasible model is formulated or when a solution is found where at least one of such domain bounds is active. The implementation of this scheme including a MATLAB-GAMS interface is depicted in Figure 7. In every iteration of the external loop,  $I_S$  sampling bounds are used in the calculation of the surrogate adjustable parameters. At this point, a MATLAB-GAMS interface calls GAMS to solve the superstructure MINLP model (i.e., Superstructure.gms) using the recently calculated surrogate parameter values, and retrieves solution information. This information includes the solution status that indicates if the problem is feasible or not, as well as details about active-inactive, feasible-infeasible constraints. This information is used by the  $I_S$  domain updater to either declare that a solution has been found or to update the domain in order to start a new iteration. The proposed scheme can also be extended to obtain more accurate surrogates over a reduced domain, thereby partially addressing potential mismatches between detailed unit models and surrogates.

In general, guaranteeing that a surrogate-based approach will converge to the optimal solution of the original model is a difficult task. Even in the special case where both surrogate and detailed models are NLP formulations, checking the sufficient conditions is rather hard, whereas ensuring that necessary conditions are satisfied can be of little use in practice (see Biegler et al.<sup>70</sup> for a theoretical discussion of such conditions in the context of optimization using approximated models). An alternative to check directly such conditions has been proposed by Caballero and Grossmann.<sup>48</sup> In their work, the surrogate domain updating procedure enters a contraction step every time the optimization step returns a feasible solution strictly interior to the surrogate domain. The idea of such contraction of the domain is to increase the local accuracy of the surrogate, which increases the chances of identifying a solution to the original detailed optimization problem.

### Plant subsystem surrogates

The use of surrogates at the unit operation level is intuitive because unit models are the natural components of any process synthesis exercise and the building blocks of com-



**Figure 7. External loop to handle the update of surrogate applicability bounds.**

[Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]

mercial simulators. However, surrogates can also be used to describe the behavior of entire plant subsystems composed by several unit operations. This is convenient when the structure of such subsystems is not under investigation; for example, when the selection of one unit operation implies the selection of another unit. The obvious advantage is the reduction in the total number of connecting variables (e.g., state variables of the streams connecting the units) and their associated surrogate expressions (in the case of the dependent variables).

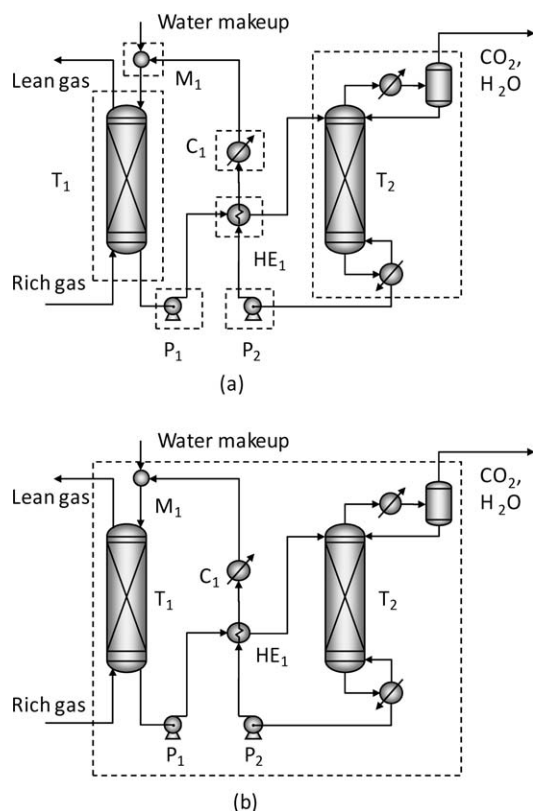
To illustrate this idea, consider the amine-based CO<sub>2</sub> capture system presented in Figure 8. Assuming five components (O<sub>2</sub>, N<sub>2</sub>, CO, H<sub>2</sub>O, MEA) and using the standard surrogate models presented in Table 1, Table 2 and Table 3, and adding capital and operational costs to the list of dependent variables in each unit, the total number of surrogate independent and dependent variables to model the systems as presented in Figure 8(a) approaches 100. However, considering a single surrogate for the entire system as presented in Figure 8(b) brings this number close to 30. This reduction is the result of: (a) some streams becoming internal to the system, and (b) the reduction in the total number of degrees of freedom (and hence the effective number of independent variables) because of unit coupling.

The construction of such plant subsystem surrogate requires the implementation of a simulation model where some of the original degrees of freedom for the individual units are related to each other; for example, the pressure increase of the pumps are related to column operating pressures and the pressure drops of the units in the streams connecting the columns. Hence, these variables are no longer independent variables, whereas some other variables are replaced by variables which are introduced to better describe the impact of the operational conditions on the system performance; for example, the total solvent flow rate and the regenerator CO<sub>2</sub> degree of separation are replaced by the absorber CO<sub>2</sub> recovery and the lean solvent CO<sub>2</sub> concentration as suggested in the literature.<sup>71</sup>

### Integration of multiplatform models

Another interesting aspect of the proposed framework is that it enables the formulation of optimization models that





**Figure 8. Amine-based CO<sub>2</sub> capture unit.**

(a) Use of surrogate models at the unit operation level ( $|I_S| + |D_S| = 99$ ), (b) Use of a surrogate for the whole system ( $|I_S| + |D_S| = 33$ ).

combine unit models developed using different tools. This is possible because models developed in different commercial simulators, programming languages, and modeling environments can be used as black boxes to generate data that is then used to fit surrogate unit models which can be easily combined. Thus, the proposed framework allows us to optimize processes that include new technologies or, in general, unit operations that cannot be readily modeled using commercial tools.

For example, consider the production of methanol from water and CO<sub>2</sub>, where the latter is split into CO and O<sub>2</sub> using a thermochemical splitter. The resulting CO is then mixed with water in a water gas shift (WGS) reactor to produce a mixture of CO, CO<sub>2</sub>, and H<sub>2</sub> that is finally feed to a methanol synthesis (MS) reactor to generate the product. Scientists from Sandia National Labs are currently exploring this pathway to methanol using a recently developed thermosplitter, called CR5.<sup>72</sup> Clearly, to optimize the process as a whole, we need to account simultaneously for the CR5 splitter and all remaining units. Although the WGS and MS reactors and separation units can be modeled using ASPEN, the CR5 reactor is too complicated: it features a configuration where multiple parallel discs supporting a catalytic material rotate inside the device, passing from an oxidation section, where CO<sub>2</sub> is stripped of one oxygen atom, to a reduction section where the catalyst is regenerated and O<sub>2</sub> is released. However, a FORTRAN model was recently developed that accounts for the effects of ring rotation, reaction

and diffusion in the active ring material in both the oxidation and reduction sections of the ring, and recuperation energy transfer with a counter-rotating ring outside the oxidation and reduction sections. A shrinking core particle model is used to account for the reaction and diffusion processes occurring on the scale of individual particles, and reacting particles are embedded within an inert solid matrix through which reacting species and heat diffuse.<sup>73</sup> Therefore, using this FORTRAN model, we can obtain results that allow us to develop a surrogate model for the CR5 (see Figure 9) that can then be combined with surrogates for the remaining units (developed using other codes or any commercial process simulator) for plant-wide optimization.

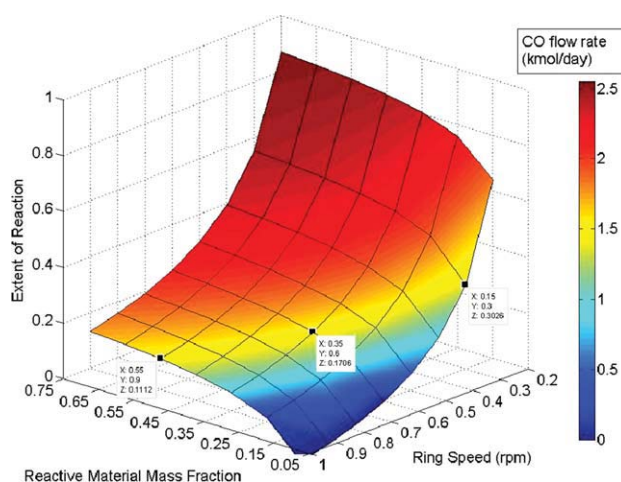
### Superstructure modeling implementation

The optimization models presented in this article were written in GAMS using compact equations blocks per equipment type, and network connectivity using stream-equipment incidence sets. As shown in Figure 7, the use of proper interfacing between ASPEN PLUS and GAMS, and MATLAB and GAMS (see MATLAB and GAMS: Interfacing Optimization and Visualization Software, by Michael C. Ferris; <http://www.cs.wisc.edu/math-prog/matlab.html>) allows the automatic information transfer between the major calculation operations, a prerequisite to make the methodology practical.

### Applications

#### Example #3: Solvent regeneration unit

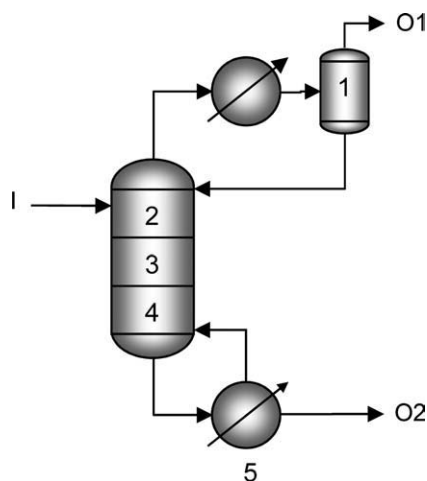
Consider the solvent regeneration column presented in Figure 10 for an amine-based carbon capture system. The system has five components (O<sub>2</sub>, N<sub>2</sub>, CO<sub>2</sub>, H<sub>2</sub>O and MEA); that is, NC = 5. The column has one feed, one partial condenser with total reflux, one partial reboiler, and five equilibrium stages (N = 5). A rigorous MESH model for a simple isobaric distillation column like this one has  $N(3NC + 5) +$



**Figure 9. CO production and extent of reaction in the CR5 subsystem model as a function of rotation ring speed and fraction of reactive material.**

[Color figure can be viewed in the online issue, which is available at [www.interscience.wiley.com](http://www.interscience.wiley.com).]





**Figure 10. Amine regeneration column.**

2 equations (including the summation of mol fractions for the feed stream as well as the thermo equations to calculate  $K$ -values and stream molar enthalpies) and  $N(3NC + 5) + (NC + 4) + 3$  variables (including the  $K$ -values and molar enthalpies for every stage and every stream), leading to  $NC + 5$  degrees of freedom. This leads to a total of 102 equations and 112 variables, 10 of which are independent (e.g., 5 feed component flows, feed temperature, feed pressure, column pressure drop, condenser temperature, and boil up ratio).

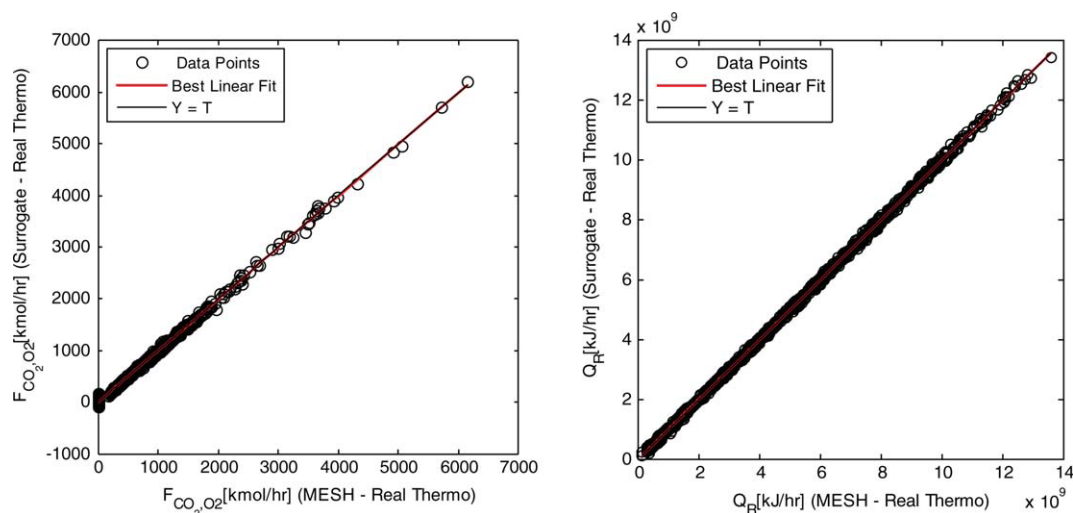
An approach that is often used to generate computationally tractable methods is to use simplified unit models. For example, in the case of distillation columns, a popular simplification comes with the use of ideal thermodynamics, where the  $K$ -values are independent of composition, and mixing effects are not considered in enthalpy calculations. Although these models can be useful in addressing simple ideal problems, their quality in general can be rather poor (e.g., ideal models cannot be used to describe amine regeneration, a case where the mixtures are highly nonideal). Our

approach does not suffer from this limitation because regardless of the complexity of the unit models used to generate unit surrogates, the form of the surrogates is exactly the same and their size is similar. Most importantly, surrogate models generated using detailed unit models can be substantially less complex and more accurate than unit models using simplified thermodynamics.

To illustrate this point, we compare a MESH model with ideal thermodynamics against a surrogate model for the column in Figure 10. Considering a fixed number of trays, a standard surrogate model for this column in the form presented in Table 3 will include the same 10 independent variables of the rigorous model and 16 expressions to calculate the connecting dependent variables (outlet stream state variables, plus condenser, and reboiler duties). The surrogate was built considering the  $I_s$  domain presented in Table 1.

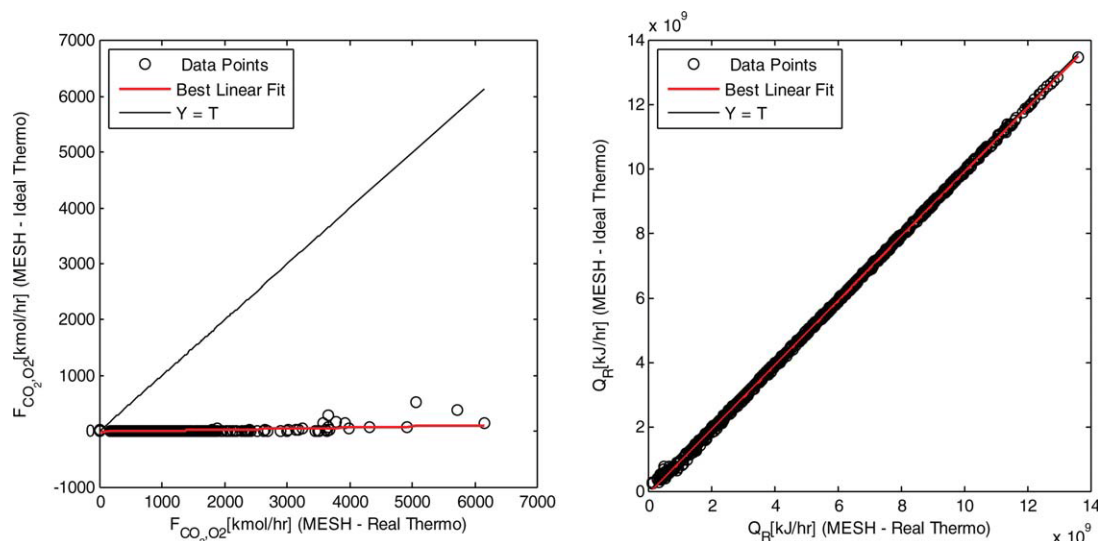
The multivariable mapping of the surrogate is an ANN with two hidden layers including 15 neurons each (with the  $\tanh(\cdot)$  activation function), and an output layer with nine linear neurons (because the number of dependent variables in the multivariable mapping section of the surrogate is only 9). It was built from sample points calculated using a MESH model in ASPEN with the amine thermo package. All calculations were performed in a PC with an Intel(R) Core(TM)i7 CPU 920 @ 2.67 GHz. The generation of the 1024 sample points used took 140 s, whereas the training of the network took 144 s. Figure 11 represents the postregression analysis for two important dependent variables: reboiler duty and bottoms  $\text{CO}_2$  flow. These two variables were selected as they are related to the tower regeneration capabilities and its operational cost. As shown, the agreement between the detailed MESH model and the surrogate is very good, with a maximum deviation of about 2% of the variable range.

A similar analysis is presented in Figure 12 for a MESH model with ideal thermodynamics. As shown, the values predicted for  $\text{CO}_2$  content in the bottoms are greatly underestimated by the ideal model, whereas the deviations are not very significant in the case of the reboiler duty. This is expected as the major contribution of a nonideal amine



**Figure 11. Postregression analysis of an amine regeneration unit surrogate.**

[Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]



**Figure 12. Effect of ideal vs. real thermodynamics in a distillation column MESH model.**

[Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]

thermo calculation package is in the correct estimation of CO<sub>2</sub>-MEA interaction. The enthalpy calculation of aqueous amine solutions does not seem as challenging in cases like the one presented here, where water is abundant.

#### Example #4: Synthesis of a reaction separation—Recycle system

Consider the simple process superstructure presented in Figure 13 for the production of maleic anhydride (MA) from Benzene. It consists of five units: a CSTR ( $R_1$ ), a PFR ( $R_2$ ), a flash tank ( $F$ ), a splitter ( $S$ ), and a mixer ( $M$ ). The goal is to find the reactor type (CSTR or PFR), reactor temperature and volume, flash temperature and recycle fraction, maximizing the annualized profit, which includes revenue from MA at current price ([www.icis.com](http://www.icis.com)), capital cost as per Guthrie<sup>57</sup> and standard utility prices.

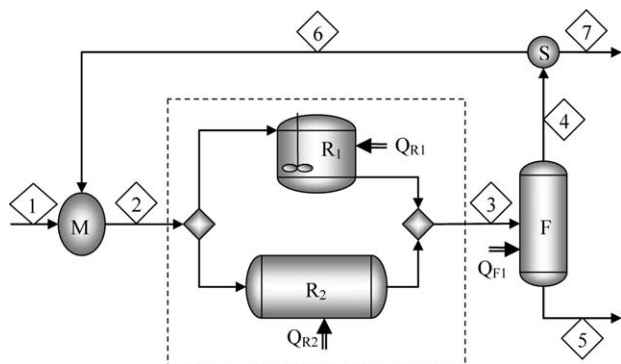
The system feed, kinetics, and annualized profit calculations are the same used previously to illustrate the construction of a surrogate model and its use in the optimal design and operation of a CSTR. The detailed models in this case

were developed in MATLAB. The formulation involves the reformulated surrogate models for the mixer ( $|I_S| = 7$ ,  $|D_S| = 7$ ), CSTR ( $|I_S| = 9$ ,  $|D_S| = 11$ ), PFR, ( $|I_S| = 9$ ,  $|D_S| = 11$ ) and flash tank ( $|I_S| = 9$ ,  $|D_S| = 15$ ). The optimization was performed using GAMS 22.5 – DICOPT. The optimal solution involves a 40 m<sup>3</sup> PFR at 680 K, a flash unit at 333 K, and recycle stream that is equal to 46% of the flash vapor stream. All calculations were performed in a PC with an Intel(R) Core(TM)i7 CPU 920 @ 2.67 GHz. The bound updating loop was performed only once. The times required for the generation of the surrogate were 82.3 s for the CSTR, 42.3 s for the PFR, 230.1 s for the flash unit, and 30.1 s for the mixer. The time required for the solution of the simplified optimization model was 4.33 s.

#### Conclusions

Although theoretically powerful, superstructure-based methodologies have not yet been widely adopted mainly due to the mathematical complexity of the resulting mathematical programming models. To address this challenge, we presented a strategy where complex unit operation models are replaced with compact yet accurate surrogate models that are generated using commercial process simulators (e.g., HYSYS, ASPEN Plus). The article discussed a variable analysis approach for the design of effective surrogate models and standard models for all common unit operations. We also presented how accurate surrogates can be generated using highly sophisticated unit models, and how superstructure representations can be formulated as MINLP models incorporating the proposed surrogate models. Finally, we discussed aspects concerning the effective solution of the resulting models.

The proposed strategy has three major advantages. First, it leads to accurate superstructure optimization models that are not prohibitively large. Most importantly, it results in unit



**Figure 13. Maleic anhydride process superstructure.**

models that are simpler and yet potentially more accurate than models currently used in superstructure approaches. Second, it offers a high-level modeling framework that allows the uniform formulation of different unit operations, thereby facilitating the study of processes whose unit models have been developed using incompatible tools. Third, it leads to formulations that involve a single type of nonlinearity, which implies that special-purpose solution algorithms can be developed to address realistic process synthesis problems. Therefore, we believe that the proposed framework lays the foundation for significant advances in the area of process synthesis.

## Notation

### Indices and sets

$c \in \mathbf{C}$  = component  
 $i \in \mathbf{I}$  = unit model variable  
 $j \in \mathbf{J}$  = unit model equation  
 $k \in \mathbf{K}$  = neural network layer or MLP layer index  
 $r \in \mathbf{R}$  = reaction  
 $s \in \mathbf{S}$  = stream  
 $u \in \mathbf{U}$  = process unit  
 $\mathbf{C}_{\text{UM}}$  = unit model connecting variables  
 $\mathbf{D}_{\text{UM}}$  = unit model dependent variables  
 $\mathbf{D}_{\text{S}}$  = surrogate model dependent variables  
 $\mathbf{F}_{\text{UM}}$  = unit model fixed variables  
 $\mathbf{I}_{\text{UM}}$  = unit model independent variables  
 $\mathbf{I}_{\text{S}}$  = surrogate model independent variables  
 $\mathbf{N}_{\text{UM}}$  = unit model natural independent variables

### Subsets

$\mathbf{D}_{\text{S}}^*$  = surrogate dependent variables calculated using the nonlinear multivariable mapping  
 $\mathbf{S}_u$  = streams of unit  $u$   
 $\mathbf{S}_{\text{I}}^u$  = inlet streams of unit  $u$   
 $\mathbf{S}_{\text{O}}^u$  = outlet streams of unit  $u$   
 $\mathbf{S}_{\text{PD}}^u$  = superstructure product streams  
 $\mathbf{S}_{\text{RM}}^u$  = superstructure raw material streams  
 $\mathbf{U}^{\text{AC}}$  = simple absorption columns  
 $\mathbf{U}^{\text{C}}$  = conditional units  
 $\mathbf{U}^{\text{CE}}$  = compression-expansion units  
 $\mathbf{U}^{\text{DC}}$  = simple distillation columns  
 $\mathbf{U}^{\text{EV}}$  = expansion valves  
 $\mathbf{U}^{\text{FV}}$  = flash vessels  
 $\mathbf{U}^{\text{HC}}$  = heater-coolers  
 $\mathbf{U}^{\text{M}}$  = stream mixers  
 $\mathbf{U}^{\text{P}}$  = permanent units  
 $\mathbf{U}^{\text{S}}$  = stream splitters  
 $\mathbf{U}^{\text{R}}$  = reactors

### Variables

$(A/P)$  (---) = inverse annuity factor  
 $Br_u$  (---) = boil-up ratio of distillation unit  $u$   
 $C_{c,s}$  (kmol/m<sup>3</sup>) = molar concentration of component  $c$  in stream  $s$   
 $CC_u$  (USD) = capital cost of unit  $u$   
 $COP_u$  (USD/yr) = annual operating cost of unit  $u$   
 $D_u$  (m) = internal diameter of reactor  $u$   
 $\Delta P_u$  (kPa) = pressure drop of unit  $u$   
 $\Delta P_u$  (kPa) = pressure increase of unit  $u$   
 $e_u$  (---) = isentropic efficiency of compression unit  $u$   
 $F_{c,s}$  (kmol/h) = molar flow rate of component  $c$  in stream  $s$   
 $h_s$  (kJ/kmol) = molar enthalpy of stream  $s$   
 $L_u$  (m) = length of reactor  $u$   
 $N_{r,u}$  (---) = number of rectifying stages in distillation column  $u$   
 $N_{s,u}$  (---) = number of stripping stages in distillation column  $u$   
 $N_{a,u}$  (---) = number of absorption stages in absorber  $u$   
 $\xi_{s,u}$  (---) = split fraction for stream  $s$  in splitter  $u$   
 $P_s$  (kPa) = pressure of stream  $s$

$Q_u$  (kJ/h) = heating duty of unit  $u$   
 $r_r$  [kmol/(m<sup>3</sup> h)] = rate of reaction  $r$   
 $T_{\text{cond},u}$  [K] = Condenser temperature of distillation unit  $u$   
 $T_s$  (K) = temperature of stream  $s$   
 $T_u$  (K) = temperature of unit  $u$   
 $\mathbf{u}^k$  = ANN outlet vector of layer  $k$   
 $v_s$  (m<sup>3</sup>/kmol) = molar volume of stream  $s$   
 $V_u$  (m<sup>3</sup>) = internal volume in reactor  $u$   
 $W_u$  (kW) = power consumption/generation of unit  $u$   
 $x_{c,s}$  (---) = mol fraction of component  $c$  in stream  $s$   
 $x_{ij}$  (---) = matching binary for variable  $z_i$  and equation  $e_j$   
 $X_{r,u}$  (kmol/h) = extent rate for reaction  $r$  in unit  $u$   
 $y_i$  (---) = selection binary for variable  $z_i$  in  $\mathbf{I}_{\text{UM}}$   
 $Y_u$  (---) = activation/deactivation Boolean variable for unit  $u$   
 $\psi_u$  = vector grouping unit  $u$  operating variables (e.g., pressure drop, heat duty and volume in the case of an isothermal CSTR)

### Parameters

$\mathbf{A} = \{\mathbf{a}_{ij}\}$  = incidence matrix of variables  $z_i$  in equations  $e_j$   
 $\mathbf{b}^k$  = ANN bias vector of layer  $k$   
 $\mathbf{U}_L^k$  = lower bound on outlet vector of layer  $k$   
 $\mathbf{U}_U^k$  = upper bound on outlet vector of layer  $k$   
 $v_{r,c}$  = stoichiometric coefficient of component  $c$  in reaction  $r$   
 $\pi_s$  (USD/kmol) = molar price of stream  $s$   
 $\phi_i$  = selection preference coefficient for variable  $z_i$   
 $\mathbf{W}^k$  = ANN weight matrix of layer  $k$

## Literature Cited

- Douglas JM. A hierarchical decision procedure for process synthesis. *AIChE J.* 1985;31:353–362.
- Douglas JM. *Conceptual Design of Chemical Processes*. New York: McGraw-Hill, 1988.
- Han C, Stephanopoulos G, Douglas J. Automation in design: the conceptual synthesis of chemical processing schemes. *Adv Chem Eng.* 1995;21:93–146.
- Li XN, Kraslawski A. Conceptual process synthesis: past and current trends. *Chem Eng Process.* 2004;43:583–594.
- Glasser D, Hildebrandt D, Crowe C. A geometric approach to steady flow reactors—the attainable region and optimization in concentration space. *Ind Eng Chem Res.* 1987;26:1803–1810.
- Zhou W, Manousiouthakis VI. Non-ideal reactor network synthesis through IDEAS: attainable region construction. *Chem Eng Sci.* 2006; 61:6936–6945.
- Widagdo S, Seider WD. Azeotropic distillation. *AIChE J.* 1996;42: 96–130.
- Fien GJAF, Liu YA. Heuristic synthesis and shortcut design of separation processes using residue curve maps—a review. *Ind Eng Chem Res.* 1994;33:2505–2522.
- Linnhoff B. Pinch analysis—a state-of-the-art overview. *Chem Eng Res Design.* 1993;71:503–522.
- Elhalwagi MM, Manousiouthakis V. Synthesis of mass exchange networks. *AIChE J.* 1989;35:1233–1244.
- Grossmann IE. Mixed-integer programming approach for the synthesis of integrated process flowsheets. *Comput Chem Eng.* 1985;9: 463–482.
- Yeomans H, Grossmann IE. A systematic modeling framework of superstructure optimization in process synthesis. *Comp Chem Eng.* 1999;23:709–731.
- Achenie LKE, Biegler LT. A superstructure based approach to chemical reactor network synthesis. *Comput Chem Eng.* 1990;14:23–40.
- Kokossis AC, Floudas CA. Optimization of complex reactor networks 2. Nonisothermal operation. *Chem Eng Sci.* 1994;49:1037–1051.
- Lakshmanan A, Biegler LT. Synthesis of optimal chemical reactor networks. *Ind Eng Chem Res.* 1996;35:1344–1353.
- Aggarwal A, Floudas CA. Synthesis of general distillation sequences—nonsharp separations. *Comput Chem Eng.* 1990;14:631–653.

17. Bauer MH, Stichlmair J. Superstructures for the mixed integer optimization of nonideal and azeotropic distillation processes. *Comput Chem Eng.* 1996;20:S25–S30.
18. Barttfeld M, Aguirre PA, Grossmann IE. A decomposition method for synthesizing complex column configurations using tray-by-tray GDP models. *Comput Chem Eng.* 2004;28:2165–2188.
19. Heckl I, Kovacs Z, Friedler F, Fanc LT, Liuc J. Algorithmic synthesis of an optimal separation network comprising separators of different classes. *Chem Eng Process.* 2007;46:656–665.
20. Papoulias SA, Grossmann IE. A structural optimization approach in process synthesis .2. Heat-recovery networks. *Comput Chem Eng.* 1983;7:707–721.
21. Floudas CA, Grossmann IE. Automatic-generation of multiperiod heat-exchanger network configurations. *Comput Chem Eng.* 1987;11:123–142.
22. Sorsak A, Kravanja Z. Simultaneous MINLP synthesis of heat exchanger networks comprising different exchanger types. *Comput Chem Eng.* 2002;26:599–615.
23. Barbaro A, Bagajewicz MJ. New rigorous one-step MILP formulation for heat exchanger network synthesis. *Comput Chem Eng.* 2005;29:1945–1976.
24. Friedler F, Tarjan K, Huang YW, Fan LT. Graph-theoretic approach to process synthesis—axioms and theorems. *Chem Eng Sci.* 1992;47:1973–1988.
25. Friedler F, Tarjan K, Huang YW, Fan LT. Graph-theoretic approach to process synthesis - polynomial algorithm for maximal structure generation. *Comput Chem Eng.* 1993;17:929–942.
26. Brendel MH, Friedler F, Fan LT. Combinatorial foundation for logical formulation in process network synthesis. *Comput Chem Eng.* 2000;24:1859–1864.
27. Bagajewicz MJ, Manousiouthakis V. Mass heat-exchange network representation of distillation networks. *AIChE J.* 1992;38:1769–1800.
28. Drake JE, Manousiouthakis V. IDEAS approach to process network synthesis: minimum plate area for complex distillation networks with fixed utility cost. *Ind Eng Chem Res.* 2002;41:4984–4992.
29. Zhou W, Manousiouthakis VI. Pollution prevention through reactor network synthesis: the IDEAS approach. *Int J Environ Pollut.* 2007;29:206–231.
30. Kocis GR, Grossmann IE. A modeling and decomposition strategy for the MINLP optimization of process flowsheets. *Comput Chem Eng.* 1989;13:797–819.
31. Daichendt MM, Grossmann IE. Integration of hierarchical decomposition and mathematical programming for the synthesis of process flowsheets. *Comput Chem Eng.* 1998;22:147–175.
32. Kravanja Z, Grossmann IE. Multilevel-hierarchical MINLP synthesis of process flowsheets. *Comput Chem Eng.* 1997;21:S421–S426.
33. Bedenik NI, Pahor B, Kravanja Z. An integrated strategy for the hierarchical multilevel MINLP synthesis of overall process flowsheets using the combined synthesis/analysis approach. *Comput Chem Eng.* 2004;28:693–706.
34. Shao T, Krishnamurty S, Wilmes GC. Preference-based surrogate Modeling in engineering design. *AIAA J.* 2007;45:2688–2701.
35. Xiong Y, Chen W, Apley D, Ding X. A non-stationary covariance-based Kriging method for metamodeling in engineering design. *Int J Numer Meth Eng.* 2007;71:733–756.
36. Won KS, Ray T. A framework for design optimization using surrogates. *Eng Optim.* 2005;37:685–703.
37. Jones DR, Schonlau M, Welch WJ. Efficient global optimization of expensive black-box functions. *J Global Optim.* 1998;13:455–492.
38. Lucia A. Multi-scale methods and complex processes: a survey and look ahead. In *Proceedings of the Seventh International Conference on the Foundations of Computer-aided Process Design*. Breckenridge, CO: CRC Press, 2009.
39. Haykin, S. *Artificial Neural Networks: A comprehensive foundation*. Upper Saddle River, NJ: Prentice Hall, 1999.
40. Cressie NA. *Statistics for Spatial Data*. New York: Wiley-Interscience, 1993.
41. Meert K, Rijckaert M. Intelligent modelling in the chemical process industry with neural networks: a case study. *Comput Chem Eng.* 1998;22:S587–S593.
42. Bloch G, Denoeux T. Neural networks for process control and optimization: two industrial applications. *ISA Trans.* 2003;42:39–51.
43. Fernandes FAN. Optimization of Fischer-Tropsch synthesis using neural networks. *Chem Eng Technol.* 2006;29:449–453.
44. Mujtaba IM, Aziz N, Hussain MA. Neural network based modelling and control in batch reactor. *Chem Eng Res Design.* 2006;84:635–644.
45. Palmer K, Realff M. Metamodeling approach to optimization of steady-state flowsheet simulations—model generation. *Chem Eng Res Design.* 2002;80:760–772.
46. Palmer K, Realff M. Optimization and validation of steady-state flowsheet simulation metamodels. *Chem Eng Res Design.* 2002;80:773–782.
47. Davis E, Ierapetritou M. A kriging method for the solution of non-linear programs with black-box functions. *AIChE J.* 2007;53:2001–2012.
48. Caballero JA, Grossmann IE. An algorithm for the use of surrogate models in modular flowsheet optimization. *AIChE J.* 2008;54:2633–2650.
49. Bunus P, Fritzson P. Automated static analysis of equation-based components. *Simulation—Trans Soc Model Simul Int.* 2004;80:321–345.
50. Mckay MD, Beckman RJ, Conover WJ. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics.* 2000;42:55–61.
51. Pukelsheim F. *Optimal Design of Experiments*. Classic ed. Philadelphia: SIAM/Society for Industrial and Applied Mathematics, 2006.
52. *ASPENplus Input Language Guide*. Aspen Technology, Inc., v. 7.0, July 2008.
53. Biegler LT, Grossmann IE, Westerberg AW. *Systematic Methods of Chemical Process Design*. Upper Saddle River, N.J.: Prentice Hall PTR, 1997.
54. Demuth H, Beale M, Hagan M. *MATLAB Neural Network Toolbox™* 6. Natick: The MathWorks, Inc, 2009.
55. Nguyen D, Widrow B. Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. *Proc Int Joint Conf Neural Networks.* 1990;3:21–26.
56. Westerink EJ, Westerterp KR. Safe design of cooled tubular reactors for exothermic multiple reactions—multiple-reaction networks. *Chem Eng Sci.* 1988;43:1051–1069.
57. Guthrie KM. Data and techniques for preliminary capital cost estimating. *Chem Eng.* 1969;76:114–142.
58. Baumrucker BT, Renfro JG, Biegler LT. MPEC problem formulations and solution strategies with chemical engineering applications. *Comput Chem Eng.* 2008;32:2903–2913.
59. Raghunathan AU, Diaz MS, Biegler LT. An MPEC formulation for dynamic optimization of distillation operations. *Comput Chem Eng.* 2004;28:2037–2052.
60. Ferris MC, Dirkse SP, Jagla JH, Meeraus A. An extended mathematical programming framework. *Comput Chem Eng.* 2009;33:1973–1982.
61. Raman R, Grossmann IE. Modeling and computational techniques for logic-based integer programming. *Comput Chem Eng.* 1994;18:563–578.
62. Raman R, Grossmann IE. Symbolic-integration of logic in mixed-integer linear-programming techniques for process synthesis. *Comput Chem Eng.* 1993;17:909–927.
63. Vecchiotti A, Grossmann IE. LOGMIP: a disjunctive 0-1 non-linear optimizer for process system models. *Comput Chem Eng.* 1999;23:555–565.
64. Grossmann IE, Lee S. Generalized convex disjunctive programming: nonlinear convex hull relaxation. *Comput Optim Appl.* 2003;26:83–100.
65. Sawaya NW, Grossmann IE. Computational implementation of non-linear convex hull reformulation. *Comput Chem Eng.* 2007;31:856–866.
66. Balas E. Disjunctive programming and a hierarchy of relaxations for discrete optimization problems. *Siam J Algebraic Discrete Methods.* 1985;6:466–486.
67. Stubbs RA, Mehrotra S. A branch-and-cut method for 0-1 mixed convex programming. *Math Program.* 1999;86:515–532.
68. Porn R, Harjunkoski I, Westerlund T. Convexification of different classes of non-convex MINLP problems. *Comput Chem Eng.* 1999;23:439–448.



69. Bjork KM, Lindberg PO, Westerlund T. Some convexifications in global optimization of problems containing signomial terms. *Comput Chem Eng.* 2003;27:669–679.
70. Biegler LT, Grossmann IE, Westerberg AW. A note on approximation techniques used for process optimization. *Comput Chem Eng.* 1985;9:201–206.
71. Alie C, Backham L, Croiset E, Douglas PL. Simulation of CO<sub>2</sub> capture using MEA scrubbing: a flowsheet decomposition method. *Energy Convers Manag.* 2005;46:475–487.
72. Diver RB, Miller JE, Allendorf MD, Siegel N, Hogan R. Solar thermochemical water-splitting ferrite-cycle heat engines. *J Sol Energ Eng-Trans Asme* 2008;130:1001–1008.
73. Evans G. *S2P CRS Modeling for System Studies*, Albuquerque, NM: Sandia National Laboratories, 2010.
- Manuscript received Nov. 23, 2009, revision received Mar. 7, 2010, and final revision received June 18, 2010.*
-