Translation of Variables and Implementation of Efficient Logic-Based Techniques in the MINLP Process Synthesizer MIPSYN

Marcel Ropotar
Faculty of Chemistry and Chemical Engineering, University of Maribor, Maribor, SI 2000, Slovenia
Tanin Sevnica kemična industrija d.d., Sevnica, SI 8290, Slovenija

Zdravko Kravanja
Faculty of Chemistry and Chemical Engineering, University of Maribor, Maribor, SI 2000, Slovenia

DOI 10.1002/aic.11916
Published online August 26, 2009 in Wiley InterScience (www.interscience.wiley.com).

This article describes alternative GDP formulation and convex hull representations for process synthesis problems and their implementation in a unique MINLP process synthesizer MIPSYN. A special translation of variables in mixed-integer, relaxed, and logic-based variations has been proposed, which enables modeling and solving process alternatives in a narrowed lifted space of variables, defined by nonzero lower and upper bounds. Based on these translation variations, alternative formulations have been developed for convex hulls, multiple-term generalized disjunctive programming problems, and logic-based outer-approximation algorithm, all of them being specialized for the synthesis of process flowsheets. Several studies were performed and three different large-scale synthesis problems were solved to test the performance and efficiency of different formulations. This initial research indicates that the proposed alternative convex hull representation usually outperforms the conventional one when solving both MILP and NLP steps in highly combinatorial MINLP process networks problems. © 2009 American Institute of Chemical Engineers *AIChE J,* 55: 2896–2913, 2009
Keywords: generalized disjunctive programming, process synthesis, MINLP, convex hull, translation of variables, discrete-continuous translation of variables, mixed-integer translation of variables, logic-based translation of variables

Introduction

Over the last couple of decades, significant advances have been achieved in modeling and developing mathematical programming techniques (e.g., Grossmann and Kravanja[1]; Biegler and Grossmann[2]). Recent developments in logic-based optimization (e.g., Grossmann and Biegler[3]) are regarded as one of the most important achievements for effective modeling and solving of discrete-continuous synthesis problems. One of the possible representations of discrete-continuous problems is Generalized Disjunctive Programming (GDP), developed by Raman and Grossmann[4] as an extension of the disjunctive programming paradigm developed by Balas.[5] GDP was first applied to problems formulated in equation oriented environment, e.g., in the optimal design of reactive distillation columns[6] and retrofit design.[7] It was also applied to batch processes, e.g., in the synthesis of biotechnological processes,[8] and finally to problems formulated by implicit models in modular

Correspondence concerning this article should be addressed to Z. Kravanja at zdravko.kravanja@uni-mb.si

process simulators, e.g., the rigorous design of distillation columns,[9] and flowsheet optimization with complex cost and design functions.[10]

GDP problems can be solved either by transforming them into mixed-integer programs or through the development of specific solution methods, e.g., the branch and bound algorithm with convex relaxation by Lee and Grossmann.[11] Initially, disjunctions were represented by the big-M formulation,[4] while they were later on reformulated by a much more efficient convex hull representation.[11–13] Stein et al.[14] investigated the theoretical properties of different continuous reformulations in discrete-continuous optimization problems with regard to their numerical solution. Lee and Grossmann[15] reviewed the solution strategies for GDP models and also discussed the global optimization of nonconvex GDP problems,[16,17] which has become an important topic in recent years. To circumvent division by zero in nonlinear convex relaxation problems while preserving the convex nature of the problem, Sawaya and Grossmann[18] proposed two different sets of convex constraints. Sawaya[19] showed that the tightness of the convex hull representation could be significantly improved by intersecting the disjunctive sets previous to the relaxation of the linear GDPs.

It should be noted that the reformulation with a convex hull is usually tighter than the big-M one, but it has a bigger problem size.[20] To avoid this increase in problem size while keeping a tight formulation, the cutting plane method was proposed.[19,21,22]

We will describe later another attempt to increase the efficiency of GDP. An alternative convex hull representation (ACH) is proposed based on the translation of variables. The motivation for the expected improvement of efficiency relies on the following assumptions:

i) Narrowing of the variables' feasible space: In discrete-continuous problems, when an alternative is selected, variables $\mathbf{x}$ are usually defined between nonzero lower and upper bounds, $\mathbf{x}^{LO} \leq \mathbf{x} \leq \mathbf{x}^{UP}$. However, when the alternative is rejected, the corresponding bounding logical relations (e.g., $\mathbf{x}LO \cdot y \leq \mathbf{x} \leq \mathbf{x}UP \cdot y$) force their values to zero. Hence, their feasible ranges have to be enlarged to $0 \leq \mathbf{x} \leq \mathbf{x}^{UP}$. Instead of the conventional convex hull (CCH) reformulation of GDP, where the continuous variables $\mathbf{x}$ are forced to zero values when the corresponding disjunctives are false, a new formulation is proposed where the variables are forced into arbitrarily-forced values, $\mathbf{x}^{f}$, defined between the original nonzero lower and upper bounds ($\mathbf{x}^{LO} \leq \mathbf{x}^{f} \leq \mathbf{x}^{UP}$). In this way, the original space of the variable is preserved irrespective of the discrete decisions. Intuitively, one may expect that the retaining in the narrowed lifted space of variables would increase the efficiency of the GDP.

ii) Reduction of the model size: As will be shown later, depending on the selection of $\mathbf{x}^{f}$ some bounding mixed-integer logical constraints may be reduced to pure lower and upper bounds. Another reduction of constraints is achieved from applying them only to a subset of those local variables, through which alternatives are related to global constraints and the objective function.

It should be noted that although several general-purpose MINLP solvers,[23] including the logic-based solver LOGMIP,[24] have been developed, almost no automated synthesis environment, based on recent advanced techniques and specializing in the synthesis of process flowsheets, has been developed so far. Therefore, this article reports the experiences gained in developing such a synthesis environment.

The article is organized in the following way: first, a multiple term GDP formulation, specialized for process synthesis, is defined and then its alternative convex nonlinear relaxation is proposed based on the introduced translation of variables. The variable translation is finally applied to the logic-based OA algorithm, where the conventional OA master problem and its converted MILP master problem are translated into alternative formulations. A general multiple term disjunctive model formulation for typical synthesis problems in chemical engineering is then defined, based on the alternative convex hull formulation applied to the process superstructure model. Different model formulations with big-M, convex hull and alternative convex hull representations are illustrated through a small problem and then compared by solving some nontrivial synthesis examples. We report our experience in the implementation of the different formulations in the MINLP process synthesizer MIPSYN (Mixed-Integer Process SYNthesizer), the successor of PROSYN-MINLP,[25] and discuss their features as regards their mathematical structures and numerical solutions.

The Alternative Convex Hull Representation

This chapter describes the alternative convex hull formulation and the alternative logic-based outer-approximation method.

GDP problems

An alternative way of formulating discrete-continuous nonlinear problems is to use Generalized Disjunctive Programming—GDP[4] in the form:

$$
\begin{aligned}
&\min Z = \sum_k \sum_i c_{ik} + f(\mathbf{x}) \\
&\text{s.t.} \qquad \mathbf{h}^{g}(\mathbf{x}) \leq 0 \\
&\qquad\quad \mathbf{A}^{g}(\mathbf{x}) \leq \mathbf{b}^{g} \\
&\bigvee_{i \in D_k} \begin{bmatrix} Y_{ik} \\ h_{ik}(\mathbf{x}) \leq 0 \\ A^{ik}(\mathbf{x}) \leq b^{ik} \\ c_{ik} = \gamma_{ik} \end{bmatrix}, k \in \mathrm{SD} \qquad \text{(GDP)} \\
&\Omega(\mathbf{Y}) = \text{True} \\
&\mathbf{x} \in \mathbb{R}^{n},\ c \in \mathbb{R}^{m},\ \mathbf{Y} \in \{\text{True}, \text{False}\}^{m}
\end{aligned}
$$

where qualitative logical and discrete decisions are represented by disjunctives $k \in \mathrm{SD}$, which can have several terms $i \in D_k$ and propositional logical constraints $\Omega(Y)$, whilst continuous quantitative decisions are represented by (non)linear (in)equality constraints, which can be global ($\mathbf{h}^{g}(\mathbf{x}) \leq 0$, $\mathbf{A}^{g}(\mathbf{x}) \leq \mathbf{b}^{g}$) or belong to local representations of alternatives ($h_{ik}(\mathbf{x}) \leq 0$, $A^{ik}(\mathbf{x}) \leq b^{ik}$). Note that when an alternative is not selected in the (GDP), its Boolean variable Y_{ik} is false and its constraints are not applied.

For process network problems, Turkay and Grossmann[26] proposed GDP that has only two terms in each disjunction to denote the selection (Y_k is true) or rejection (Y_k is false) of process units:

$$
\begin{aligned}
&\min Z = \sum_k c_k + f(\mathbf{x}) \\
&\text{s.t.} \qquad \mathbf{h}^{\mathrm{g}}(\mathbf{x}) \le \mathbf{0} \\
&\qquad\qquad \mathbf{A}^{\mathrm{g}}(\mathbf{x}) \le \mathbf{b}^{\mathrm{g}} \\
&\begin{bmatrix} Y_k \\ h_k(\mathbf{x}) \le 0 \\ A^k(\mathbf{x}) \le b^k \\ c_k = \gamma_k \end{bmatrix} \vee \begin{bmatrix} \neg Y_k \\ B^k \mathbf{x} = 0 \\ c_k = 0 \end{bmatrix}, k \in \mathrm{SD} \qquad \text{(2T-GDP)} \\
&\Omega(Y) = \text{True} \\
&\mathbf{x} \in \mathbb{R}^n,\ \mathbf{c} \in \mathbb{R}^m,\ \mathbf{Y} \in \{\text{True, False}\}^m
\end{aligned}
$$

To expand a typical structure of process flowsheet, which consists of interconnection nodes (multiple choice mixers and splitters) and process unit nodes (Figure 1), first described in the introduction of the Modeling and Decomposition strategy (M/D) by Kocis and Grossmann,[27] an extension of the two term GDP (2T-GDP) to a multiple term GDP is proposed:

$$
\begin{aligned}
&\min Z = \sum_k \sum_i \left(c_{ik} + \alpha_{ik}^{\mathrm{a}}\right) + f^{\mathrm{g}}(\mathbf{x}^{\mathrm{g}}) \\
&\text{s.t.} \qquad \mathbf{h}^{\mathrm{g}}(\mathbf{x}^{\mathrm{g}}) \le 0 \\
&\qquad\quad \mathbf{A}^{\mathrm{g}}(\mathbf{x}^{\mathrm{g}}) \le \mathbf{b}^{\mathrm{g}} \\
&\qquad\quad \mathbf{A}^{\mathrm{r}}(\mathbf{x}^{\mathrm{g}}, \mathbf{x}^{\mathrm{s}}) \le \mathbf{b}^{\mathrm{r}} \\
&\left.\begin{array}{c}
\underset{i \in D_k}{\vee} \begin{bmatrix} Y_{ik} \\ h_{ik}(\mathbf{x}^{\mathrm{s}}) \le 0 \\ A^{ik}(\mathbf{x}^{\mathrm{s}}) \le b^{ik} \\ c_{ik} = \gamma_{ik} \\ \alpha_{ik}^{\mathrm{a}} = f_{ik}^{\mathrm{a}}(x^{\mathrm{s}}) \end{bmatrix} \\
\vee_{ik} \\
\begin{bmatrix} \neg Y_{ik} \\ B^{ik}\mathbf{x}^{\mathrm{s}} = \mathbf{0} \\ c_{ik} = 0 \\ \alpha_{ik}^{\mathrm{a}} = 0 \end{bmatrix}
\end{array}\right\} k \in \mathrm{SD} \qquad \text{(MT-GDP)} \\
&\qquad\qquad \Omega(Y) = \text{True} \\
&\mathbf{x} = (\mathbf{x}^{\mathrm{g}}, \mathbf{x}^{\mathrm{s}}) \in \mathbb{R}^n,\ \mathbf{c}, \alpha_{ik}^{\mathrm{a}} \in \mathbb{R}^m,\ \mathbf{Y} \in \{\text{True, False}\}^m
\end{aligned}
$$

where disjunctives $k \in \mathrm{SD}$ typically correspond to pairs of single-choice splitters and mixers, and terms of the disjunction $i \in D_k$ to process unit alternatives. Parallel and series arrangements as well as by-passes can appear as additional terms in disjunctions, as in Figure 1. Note that an objective term $f(\mathbf{x})$ is decomposed into a global term f^{g} $(\mathbf{x}^{\mathrm{g}})$ and terms belonging to alternatives, α_{ik}^{a}. Note also that in (MT-GDP), only one term i from D_k in each disjunction $k \in \mathrm{SD}$ can be chosen, while zero constraints (B^{ik} $(\mathbf{x}^{\mathrm{s}}) \le \mathbf{0}$; $c_{ik} = 0$; $\alpha_{ik}^{\mathrm{a}} = 0$) are applied for the rest of them; e.g., in Figure 1a, either PU1 or PU2 or (PU3 parallel to PU4) from D_1 can be selected first and then rejections of the nonselected units are chosen at the second level (Figure 1b). This is useful if variables describing nonexisting process units have to obtain zero values (e.g., flows, fixed costs), and if disjunctions have more than two terms. Therefore, problem (MT-GDP) can be viewed as a two-dimensional disjunctive problem: alternative process units are arranged along the first disjunctive dimension, and their selection/rejection along the second dimension. Note that constraints $\mathbf{A}^{\mathrm{r}}$ $(\mathbf{x}^{\mathrm{g}}, \mathbf{x}^{\mathrm{s}}) \le b^{\mathrm{r}}$ are inserted to relate global and local parts of the model where $\mathbf{x}^{\mathrm{g}}$ is a vector of global variables and $\mathbf{x}^{\mathrm{s}}$ a vector of local variables. In process flowsheet problems, these constraints typically correspond to interconnection nodes, heat integration, and other system models. Let us first present a convex hull formulation in a narrowed lifted space of variables.

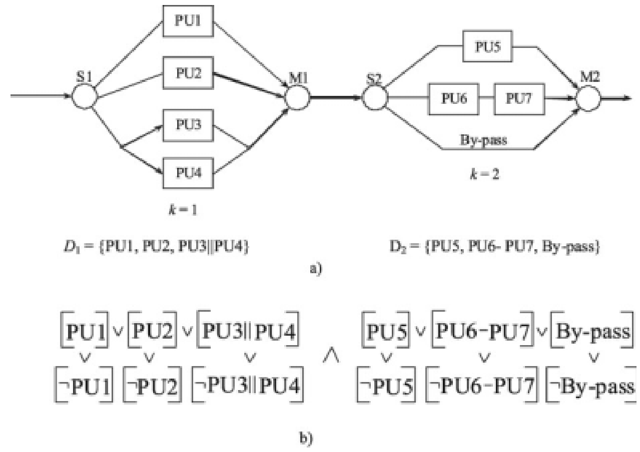

Figure 1. (a) A process superstructure consisting of interconnection and process unit nodes. (b) Logical relationship between alternatives.

Variable translation in convex hull

To define (MT-GDP) and the related logic-based OA in the narrowed lifted space of local variables, $x^{\mathrm{LO}} \le x^{\mathrm{a}} \le x^{\mathrm{UP}}$, rather than in the zero-lower-bounded one $0 \le x^{\mathrm{s}} \le x^{\mathrm{UP}}$, a special discrete-continuous translation of variables will be introduced. It will first be applied to the convex hull relaxation of the problem (MT-GDP) and then to the logic-based OA method.

The main idea of the translation is to substitute zero-lower-bounded variables x^{s} of alternatives ($0 \le x^{\mathrm{s}} \le x^{\mathrm{UP}}$) with a nonzero-lower-bounded variable x^{a} defined between the bounds ($x^{\mathrm{LO}} \le x^{\mathrm{a}} \le x^{\mathrm{UP}}$) using the following equation:

$$x^{\mathrm{s}} = x^{\mathrm{a}} - x^{\mathrm{f}}(1 - y) \qquad (1)$$

where x^{f} is an arbitrarily-forced value, and y is a corresponding binary variable. When an alternative is selected, an integer term $x^{\mathrm{f}}(1 - y)$ becomes zero and x^{s} becomes equal to x^{a}, and when it is rejected, a value x^{f} is subtracted from the variable x^{a}. If a binary variable y is relaxed to a continuous variable λ defined between 0 and 1, the following relaxed translation formula is obtained:

$$x^{\mathrm{s}} = x^{\mathrm{a}} - x^{\mathrm{f}}(1 - \lambda) \qquad (2)$$

A convex hull representation is the tightest relaxation of disjunctions k in the above problems. It is generated by means of taking the linear combination of all the points in the feasible regions of disjunctions. By applying the convex hull relaxation to the multiple term problem (MT-GDP) in the spirit of Lee and Grossmann,[11] the (CHRP) problem given below is obtained. When its zero-lower-bounded variables are translated into nonzero-lower-bounded variables using the translation Eq. 2, the following alternative (A-CHRP) problem is obtained:

$$
\begin{aligned}
&\text{(CHRP):}\\
&\min Z=\sum_k\sum_i\left(\gamma_{ik}\lambda_{ik}+f_{ik}^{a}\left(\mathbf{x}_{ik}^{s}\right)\right)+f^{g}(x^{g})\\
&\text{s.t.}\quad \mathbf{h}^{g}(\mathbf{x}^{g})\le\mathbf{0}\\
&\qquad \mathbf{A}^{g}(\mathbf{x}^{g})\le\mathbf{b}^{g}\\
&1)\quad x_k^{g}=\sum_{i\in D_k}x_{ik}^{s},\quad k\in\mathrm{SD}\\
&2)\quad \sum_{i\in D_k}\lambda_{ik}=1,\ 0\le\lambda_{ik}\le1\\
&\qquad E\lambda\le e\\
&\qquad \lambda_{ik}x_{ik}^{LO}\le x_{ik}^{s},\quad i\in D_k,k\in\mathrm{SD}\\
&3)\quad x_{ik}^{s}\le\lambda_{ik}x_{ik}^{UP},\quad i\in D_k,k\in\mathrm{SD}\\
&4)\quad \lambda_{ik}g_{ik}\left(\frac{x_{ik}^{s}}{\lambda_{ik}}\right)\le0,\quad i\in D_k,k\in\mathrm{SD}\\
&5)\quad \mathbf{x}^{\mathrm{LO}}\le\mathbf{x}^{g}\le\mathbf{x}^{\mathrm{UP}}
\end{aligned}
\quad\xrightarrow{x^{s}=x^{a}-x^{f}(1-\lambda)}
$$

$$
\begin{aligned}
&\text{(A-CHRP):}\\
&\min Z=\sum_k\sum_i\left(\gamma_{ik}\lambda_{ik}+f_{ik}^{a}\left(x_{ik}^{a}-x_{ik}^{f}(1-\lambda_{ik})\right)\right)+f^{g}(\mathbf{x}^{g})\\
&\text{s.t.}\quad \mathbf{h}^{g}(\mathbf{x}^{g})\le\mathbf{0}\\
&\qquad \mathbf{A}^{g}(\mathbf{x}^{g})\le\mathbf{b}^{g}\\
&\qquad x_k^{g}=\sum_{i\in D_k}\left(x_{ik}^{a}-x_{ik}^{f}(1-\lambda_{ik})\right),\quad k\in\mathrm{SD}\\
&\qquad \sum_{i\in D_k}\lambda_{ik}=1,\ 0\le\lambda_{ik}\le1\\
&\qquad E\lambda\le e\\
&\qquad x_{ik}^{LO}\lambda_{ik}+x_{ik}^{f}(1-\lambda_{ik})\le x_{ik}^{a},\quad i\in D_k,k\in\mathrm{SD}\\
&\qquad x_{ik}^{a}\le x_{ik}^{UP}\lambda_{ik}+x_{ik}^{f}(1-\lambda_{ik}),\quad i\in D_k,k\in\mathrm{SD}\\
&\qquad \lambda_{ik}g_{ik}\left(\frac{x_{ik}^{a}-x_{ik}^{f}(1-\lambda_{ik})}{\lambda_{ik}}\right)\le0,\ i\in D_k,k\in\mathrm{SD}\\
&\qquad \mathbf{x}^{\mathrm{LO}}\le\mathbf{x}^{g}\le\mathbf{x}^{\mathrm{UP}}
\end{aligned}
$$

where x_{ik}^{s} (x_{ik}^{a}) are disaggregated variables and xLO_{ik} (xUP_{ik}) nonzero scalars in the problem (CHRP) forcing the nonzero lower (upper) bounds when an alternative is selected. A distinction should be made between superscript (LO, UP), e.g., x^{LO}, which denote bounds defined directly in the model and (LO, UP) as part of the scalars, e.g., xLO that force bounds with bounding constraints. The numbers in front of the equations indicate different types of constraints: (1) are balance constraints for disaggregated variables, (2) are relaxed exclusive-or logical constraints for variables λ_{ik}, (3) are bounding constrains for disaggregated variables, and (4) are (non)linear combination of constraints with λ_{ik} for disjunctive terms. Note that in process synthesis problems constraints of types (1)–(3) typically represent interconnection nodes while, those of type (4) define alternative units. In both convex hull representations, global variables x^{g} can in principle be defined between nonzero lower and upper bounds. However, in the conventional (CHRP) problem, disaggregated variables x^{s} should always have zero lower bounds so that they can preserve feasibility of the bounding constraints (3) when λ_{ik} becomes zero. In the case of the alternative problem (A-CHRP), when λ_{ik} takes zero value, the corresponding variables x_{ik}^{a} in the bounding constraints (3) are set to nonzero values x_{ik}^{f}. At the same time, the terms x_{ik}^{a} and x_{ik}^{f} $(1-\lambda_{ik})$ in the balance equation 1 and the objective function precisely cancel each other out, which is equivalent to obtaining zero values for x^{s} in the original problem (CHRP). Note that if the continuous variables λ_{ik} are replaced by integer variables y_{ik}, MINLP reformulation is obtained.

Example 1: Illustrative Example. The idea of the variable translation is illustrated by a subsequent simple illustrative example. The equations below represent the convex hull formulation (E1-CCH) of two feasible regions in (x, z) space. By applying the translation of variables, the following alternative convex hull formulation (E1-ACH) is obtained:

$$
\begin{aligned}
&\text{(E1-CCH):}\\
&x=x_1^{s}+x_2^{s}\\
&z=z_1^{s}+z_2^{s}\\
&xLO_1\lambda_1\le x_1^{s}\le xUP_1\lambda_1\\
&xLO_2\lambda_2\le x_2^{s}\le xUP_2\lambda_2\\
&zLO_1\lambda_1\le z_1^{s}\le zUP_1\lambda_1\\
&zLO_2\lambda_2\le z_2^{s}\le zUP_2\lambda_2\\
&\lambda_1+\lambda_2=1
\end{aligned}
\quad\xrightarrow{x^{s}=x^{a}-x^{1}(1-\lambda)}
$$

$$
\begin{aligned}
&\text{(E1-ACH):}\\
&x=x_1^{a}-x_1^{f}(1-\lambda_1)+x_2^{a}-x_2^{f}(1-\lambda_2)\\
&z=z_1^{a}-z_1^{f}(1-\lambda_1)+z_2^{a}-z_2^{f}(1-\lambda_2)\\
&x_1^{f}+\left(xLO_1-x_1^{f}\right)\lambda_1\le x_1^{a}\le x_1^{f}+\left(xUP_1-x_1^{f}\right)\lambda_1\\
&x_2^{f}+\left(xLO_2-x_2^{f}\right)\lambda_2\le x_2^{a}\le x_2^{f}+\left(xUP_2-x_2^{f}\right)\lambda_2\\
&z_1^{f}+\left(zLO_1-z_1^{f}\right)\lambda_1\le z_1^{a}\le z_1^{f}+\left(zUP_1-z_1^{f}\right)\lambda_1\\
&z_2^{f}+\left(zLO_2-z_2^{f}\right)\lambda_2\le z_2^{a}\le z_2^{f}+\left(zUP_2-z_2^{f}\right)\lambda_2\\
&\lambda_1+\lambda_2=1
\end{aligned}
$$

where x_1^{s}, z_1^{s}, x_2^{s}, z_2^{s} are zero-lower-bounded disaggregated variables, x_1^{a}, z_1^{a}, x_2^{a}, z_2^{a} are nonzero-lower-bounded disaggregated variables, and λ_1, λ_2 are continuous variables in the range $0\le\lambda_1$, $\lambda_2\le1$. For $xLO_1=0.7$, $xUP_1=1.0$, $xLO_2=0.2$, $xUP_2=0.5$, $zLO_1=10$, $zUP_1=20$, $zLO_2=40$, $zUP_2=50$, and x_1^{f}, x_2^{f}, z_1^{f}, $z_2^{f}=xLO_1$, xLO_2, zLO_1, zLO_2, convex hulls (c.f., Figure 2) are obtained for both formulations (Appendix A). Note that they are identical. They do, however, differ in the cones of their disaggregated variables: in the case of the conventional formulation (E1-CCH) the two cones have vertex points at the origin of the space (Figure 3) whilst in the case of the alternative formulation (E1-ACH), they have vertex points at the forced values (z_1^{f}, x_1^{f}) and (z_2^{f}, x_2^{f}), Figure 4. When λ is zero the disaggregated variables in the CCH formulation are forced to zero, while in the ACH formulation they are forced to x^{f}. Since forced values are always defined between the bounds, cones of alternative formulations always remain within the bounds. This allows the defining of x_i^{a} with nonzero bounds in the alternative formulation. It is interesting to note that when $x_i^{f}=xLO_i$, the first part of the type (3) bounding constraints reduces from relaxed mixed-integer constraints to simple lower bounds since the terms $(xLO_i-x_i^{f})\lambda_i$ are zero. On the other hand, the linear balanced equation 1 becomes relaxed mixed-integer constraints.

Variable translation in the logic-based OA algorithm

Lee and Grossmann[11] showed that applying the outer-approximation (OA) method to the MINLP reformulation of the convex hull relaxation of the problem (2T-GDP) gives

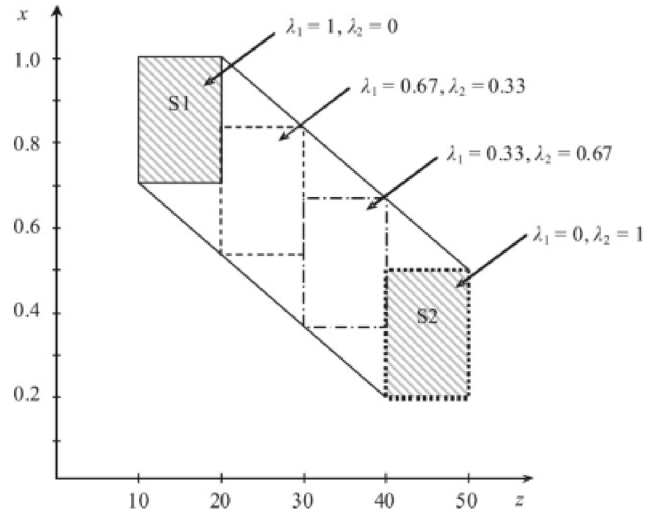

Figure 2. Identical convex hulls of feasible regions S1 and S2 for the CCH and ACH representations in space (*x*, *z*).

rise to the logic-based OA method by Turkay and Grossmann.[26] It can be shown that a similar logic-based OA method can be developed from the proposed multiple term problem (MT-GDP). Since the translation of variables is applied to it, the proposed logic-based OA algorithm is defined in the narrowed lifted space of variables. Its NLP subproblem for the fixed values of the Boolean variables at the lth iteration is given by:

$$
\begin{aligned}
&\min Z^l = \sum_{i \in D_k} \sum_{\substack{k \in \mathrm{SD,\,for} \\ Y_{ik}^l = \mathrm{True}}} \left(c_{ik} + f_{ik}^{\mathrm{a}}(\mathbf{x}^{\mathrm{a}})\right) + f^{\mathrm{g}}(\mathbf{x}^{\mathrm{g}}) \\
&\text{s.t. } \mathbf{h}^{\mathrm{g}}(\mathbf{x}^{\mathrm{g}}) \le \mathbf{0} \\
&\quad \mathbf{A}^{\mathrm{g}}(\mathbf{x}^{\mathrm{g}}) \le \mathbf{b}^{\mathrm{g}} \\
&\quad \mathbf{A}^{\mathrm{r}}(\mathbf{x}^{\mathrm{g}}, \mathbf{x}^{\mathrm{a}}) = \mathbf{b}^{\mathrm{r}} \qquad \qquad (\mathrm{NLP})^l \\
&\left.\begin{aligned}
&h_{ik}(\mathbf{x}^{\mathrm{a}}) \le 0 \\
&A^{ik}(\mathbf{x}^{\mathrm{a}}) = 0 \\
&c_{ik} = \gamma_{ik} \\
&\mathbf{x}^{\mathrm{LO}} \le \mathbf{x}^{\mathrm{a}} \le \mathbf{x}^{\mathrm{UP}} \\
&0 \le c_{ik}
\end{aligned}\right\} \forall i \in D_k, k \in \mathrm{SD} \text{ for } Y_{ik}^l = \mathrm{True}
\end{aligned}
$$

NLP subproblems are solved only for currently selected alternatives, which significantly improves the efficiency and robustness of solving discrete-continuous problems since the NLP subproblems are thus smaller and mathematical singularities are avoided when alternatives are not selected and variables take 0 values. It should be noted that even if the NLP subproblems were solved for the complete superstructure and, hence, also for rejected units, the problem of singularities would be significantly overcome, although not entirely solved, since local variables can now have only nonzero values. Note that a logic-based variation of the variable translation in Eqs. 1 and 2 can be given in the following disjunctive form:

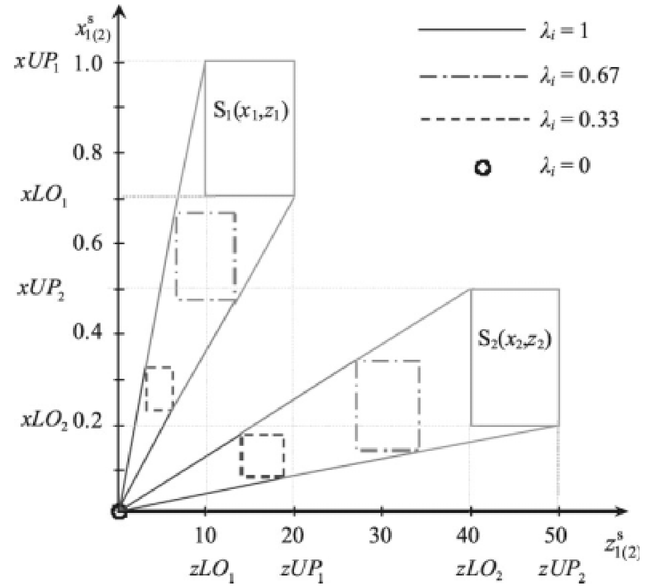

Figure 3. Cones of feasible regions S1 and S2 in space ($x_{1(2)}^{s}$, $z_{1(2)}^{s}$) for the CCH representation.

$$[Y: x^{\mathrm{s}} = x^{\mathrm{a}}] \vee [\neg Y: x^{\mathrm{s}} = x^{\mathrm{a}} - x^{\mathrm{f}}] \qquad (3)$$

where in the case of the Boolean variable $Y = \mathrm{True}$ it follows that $x^{\mathrm{s}} = x^{\mathrm{a}}$ and in the case of $Y = \mathrm{False}$ $x^{\mathrm{s}} = x^{\mathrm{a}} - x^{\mathrm{f}}$. Since NLPs are solved only for alternatives with $Y = \mathrm{True}$, only the first disjunctive $x^{\mathrm{s}} = x^{\mathrm{a}}$ is applied and the translation of variables does not affect the form of the NLP subproblems. However, it affects the form of OA master problems. The conventional OA master problem (C-OAMP) and the translated problem (A-OAMP) are given below:

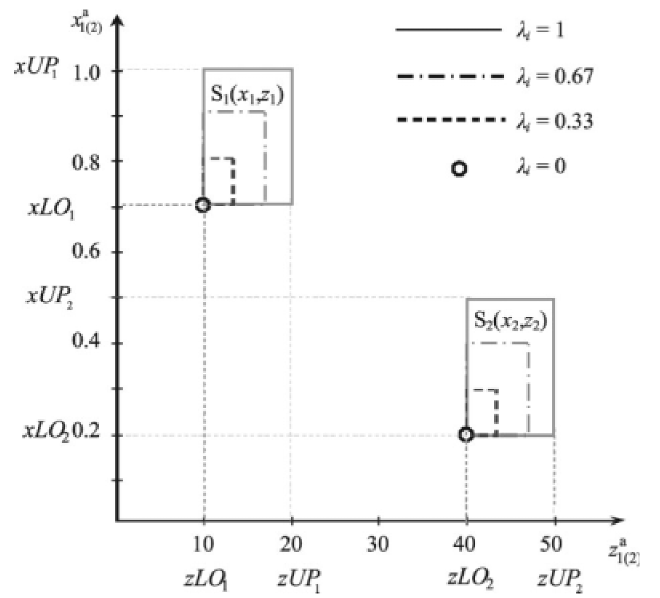

Figure 4. Cones of feasible regions S1 and S2 in space ($x_{1(2)}^{a}$, $z_{1(2)}^{a}$) for the ACH representation.

$$
\begin{array}{l}
\text{(C-OAMP):}\\
\min Z=\sum_i\sum_k\left(c_{ik}+\alpha_{ik}^{a}\right)+\alpha^{g}\\
\text{s.t.}\\
\left.\begin{array}{l}\alpha^{g}\ge f(\mathbf{x}^{l})+\nabla_x\mathbf{f}(\mathbf{x}^{l})^{T}(\mathbf{x}^{g}-\mathbf{x}^{l})\\ \mathbf{h}^{g}(\mathbf{x}^{l})+\nabla_x\mathbf{h}^{g}(\mathbf{x}^{l})^{T}(\mathbf{x}^{g}-\mathbf{x}^{l})\le 0\end{array}\right\},\ l=1,\dots,L\\
\mathbf{A}^{g}(\mathbf{x}^{g})\le\mathbf{b}^{g}\\
\mathbf{A}^{r}(\mathbf{x}^{g},\mathbf{x}^{s})\le\mathbf{b}^{r}\\
\left.\begin{array}{l}\underset{i\in D_k}{\vee}\begin{bmatrix}Y_{ik}\\ c_{ik}=\gamma_{ik}\\ \mathbf{x}^{LO}\le\mathbf{x}^{s}\le\mathbf{x}^{UP}\\ A^{ik}(\mathbf{x}^{s})\le b_{ik}\\ \alpha_{ik}^{a}\ge f_{ik}^{a}(\mathbf{x}^{l})+\nabla_x f_{ik}^{a}(\mathbf{x}^{l})^{T}(\mathbf{x}^{s}-\mathbf{x}^{l})\\ \nabla_x h_{ik}(\mathbf{x}^{l})^{T}\mathbf{x}^{s}\le\nabla_x h_{ik}(\mathbf{x}^{l})^{T}\mathbf{x}^{l}-h_{ik}(\mathbf{x}^{l}),l=1,\dots,L\end{bmatrix}\\ \vee_{ik}\begin{bmatrix}\neg\mathbf{Y}_{ik}\\ c_{ik}=0\\ \alpha_{ik}^{a}=0\\ A^{ik}\mathbf{x}^{s}=0\end{bmatrix}\end{array}\right\}k\in SD\\
\Omega(\mathbf{Y})=\text{True}\\
\mathbf{x}=(\mathbf{x}^{g},\mathbf{x}^{s})\in\mathbb{R}^{n},\ \mathbf{c}\in\mathbb{R}^{m},\ \mathbf{Y}\in\{\text{True},\text{False}\}^{m}\\
0\le\alpha^{g},\alpha_{ik}^{a},c_{ik}\\
\mathbf{x}^{LO}\le\mathbf{x}^{g}\le\mathbf{x}^{UP}
\end{array}
\quad
\xrightarrow[\left[\neg Y:x^{s}=x^{a}-x^{f}\right]]{\left[Y:x^{s}=x^{a}\right]\ \vee}
\quad
\begin{array}{l}
\text{(A-OAMP):}\\
\min Z=\sum_i\sum_k\left(c_{ik}+\alpha_{ik}^{a}\right)+\alpha^{g}\\
\text{s.t.}\\
\left.\begin{array}{l}\alpha^{g}\ge f(\mathbf{x}^{l})+\nabla_x\mathbf{f}(\mathbf{x}^{l})^{T}(\mathbf{x}^{g}-\mathbf{x}^{l})\\ \mathbf{h}^{g}(\mathbf{x}^{l})+\nabla_x\mathbf{h}^{g}(\mathbf{x}^{l})^{T}(\mathbf{x}^{g}-\mathbf{x}^{l})\le 0\end{array}\right\},\ l=1,\dots,L\\
\mathbf{A}^{g}(\mathbf{x}^{g})\le\mathbf{b}^{g}\\
\mathbf{A}^{r}(\mathbf{x}^{g},\ \Phi_{\mathbf{x}^{s}}(\mathbf{x}^{a},\mathbf{Y}))\le\mathbf{b}^{r}\\
\left.\begin{array}{l}\underset{i\in D_k}{\vee}\begin{bmatrix}Y_{ik}\\ c_{ik}=\gamma_{ik}\\ \mathbf{x}^{LO}\le\mathbf{x}^{a}\le\mathbf{x}^{UP}\\ A^{ik}(\mathbf{x}^{a})\le b_{ik}\\ \alpha_{ik}^{a}\ge f_{ik}^{a}(\mathbf{x}^{l})+\nabla_x f_{ik}^{a}(\mathbf{x}^{l})^{T}(\mathbf{x}^{a}-\mathbf{x}^{l})\\ \nabla_x h_{ik}(\mathbf{x}^{l})^{T}\mathbf{x}^{a}\le\nabla_x h_{ik}(\mathbf{x}^{l})^{T}\mathbf{x}^{l}-h_{ik}(\mathbf{x}^{l}),l=1,\dots,L\end{bmatrix}\\ \vee_{ik}\begin{bmatrix}\neg\mathbf{Y}_{ik}\\ c_{ik}=0\\ \alpha_{ik}^{a}=0\\ \mathbf{x}^{a}=\mathbf{x}^{f}\end{bmatrix}\end{array}\right\}k\in SD\\
\Omega(\mathbf{Y})=\text{True}\\
\mathbf{x}=(\mathbf{x}^{g},\mathbf{x}^{a})\in\mathbf{R}^{n},\ \mathbf{c}\in\mathbf{R}^{m},\ \mathbf{Y}\in\{\text{True},\text{False}\}^{m}\\
0\le\alpha^{g},\alpha_{ik}^{a},c_{ik}\\
\mathbf{x}^{LO}\le\mathbf{x}^{g}\le\mathbf{x}^{UP}
\end{array}
$$

where L and SD are sets of NLP solutions and disjunctions, respectively. Note that when an alternative is not selected in (C-OAMP) $\mathbf{x}^{s}$ is set to zero and its linearizations are not applied, while in (A-OAMP) $\mathbf{x}^{a}$ is set to $\mathbf{x}^{f}$, and $A^{ik}(\mathbf{x}^{a})=b_{ik}$ and linearizations are not applied. Note that in problem (A-OAMP) the translation of local variables $\mathbf{x}^{s}$ is denoted by an implicit function $\Phi_{x^{s}}(\mathbf{x}^{a},\mathbf{Y})$.

Logic-based OA problems are usually solved through MILP transformation, where Y_{ik} are replaced by binary variables y_{ik}, logical relations are formulated as integer constrains and disjunctives are represented either by a big-M or a convex hull representation. When a convex hull representation is considered, the following MILP master problem (CCH-MILP) is obtained and when it is reformulated by the mixed-integer translation of variables (Eq. 1), the following alternative MILP master problem (ACH-MILP) is obtained:

$$
\begin{array}{l}
\text{(CCH-MILP):}\\
\min\ Z=\sum_i\sum_k(c_{ik}y_{ik}+\alpha_{ik}^{a})+\alpha^{g}\\
\text{s.t.}\\
\left.\begin{array}{l}\alpha^{g}\ge f\left(\mathbf{x}^{l}\right)+\nabla_x\mathbf{f}\left(\mathbf{x}^{l}\right)^{T}(\mathbf{x}^{g}-\mathbf{x}^{l})\\ \mathbf{h}^{g}\left(\mathbf{x}^{l}\right)+\nabla_x\mathbf{h}^{g}\left(\mathbf{x}^{l}\right)^{T}(\mathbf{x}^{g}-\mathbf{x}^{l})\le 0\end{array}\right\},l=1,\dots,L\\
\mathbf{A}^{g}(\mathbf{x}^{g})\le\mathbf{b}^{g}\\
\mathbf{E}^{g}(\mathbf{y})\le\mathbf{e}^{g}\\
\mathbf{A}^{r}(\mathbf{x}^{g},\mathbf{x}^{s})\le\mathbf{b}^{r}\\
\left.\begin{array}{l}xLOy_{ik}\le\mathbf{x}^{s}\\ \mathbf{x}^{s}\le xUPy_{ik}\end{array}\right\}\quad\forall\mathbf{x}^{s}\in\mathbf{X}_{ik}\\
A^{ik}(\mathbf{x}^{s})\le b_{ik}y_{ik}\\
\nabla_x f_{ik}^{a}\left(\mathbf{x}^{l}\right)^{T}\mathbf{x}^{s}-\alpha_{ik}^{a}\le\left[\nabla_x f_{ik}^{a}\left(\mathbf{x}^{l}\right)^{T}\mathbf{x}^{l}-f_{ik}^{a}\left(\mathbf{x}^{l}\right)\right]y_{ik}\\
\nabla_x h_{ik}^{a}\left(\mathbf{x}^{l}\right)^{T}\mathbf{x}^{s}\le\left[\nabla_x h_{ik}^{a}\left(\mathbf{x}^{l}\right)^{T}\mathbf{x}^{l}-h_{ik}^{a}\left(\mathbf{x}^{l}\right)\right]y_{ik},l=1,\dots,L\\
\mathbf{x}=(\mathbf{x}^{g},\mathbf{x}^{s})\in\mathbb{R}^{n},\mathbf{y}\in\{0,1\}^{m}\\
0\le\alpha^{g},\alpha_{ik}^{a},\qquad i\in D_k,k\in SD\\
\mathbf{x}^{LO}\le\mathbf{x}^{g}\le\mathbf{x}^{UP}
\end{array}
\quad\xrightarrow{\mathbf{x}^{s}=x^{a}-\mathbf{x}^{f}(1-y)}\quad
\begin{array}{ll}
\text{(ACH-MILP):}&\\
\min\ Z=\sum_i\sum_k(c_{ik}y_{ik}+\alpha_{ik}^{a})+\alpha^{g}&\\
\text{s.t.}&\\
\left.\begin{array}{l}\alpha^{g}\ge f\left(\mathbf{x}^{l}\right)+\nabla_x\mathbf{f}\left(\mathbf{x}^{l}\right)^{T}(\mathbf{x}^{g}-\mathbf{x}^{l})\\ \mathbf{h}^{g}\left(\mathbf{x}^{l}\right)+\nabla_x\mathbf{h}^{g}\left(\mathbf{x}^{l}\right)^{T}(\mathbf{x}^{g}-\mathbf{x}^{l})\le 0\end{array}\right\},\quad l=1,\dots,L&\\
\mathbf{A}^{g}(\mathbf{x}^{g})\le\mathbf{b}^{g}&\\
\mathbf{E}^{g}(\mathbf{y})\le\mathbf{e}^{g}&\\
\mathbf{A}^{r}(\mathbf{x}^{g},\Phi_{x^{s}}(\mathbf{x}^{a},\mathbf{y}))\le\mathbf{b}^{r}&\\
\left.\begin{array}{l}x^{f}+\left(xLO-x^{f}\right)y_{ik}\le x^{a}\\ x^{a}\le x^{f}+\left(xUP-x^{f}\right)y_{ik}\end{array}\right\}\forall x^{a}\in X_{ik}&\begin{array}{r}(4)\\(5)\end{array}\\
A^{ik}\left(\mathbf{x}^{a}-\mathbf{x}^{f}(1-y_{ik})\right)\le b_{ik}y_{ik}&(6)\\
\nabla_x f_{ik}^{a}\left(\mathbf{x}^{l}\right)^{T}\mathbf{x}^{a}-\alpha_{ik}^{a}\le\nabla_x f_{ik}^{a}\left(\mathbf{x}^{l}\right)^{T}\mathbf{x}^{f}+\left[\nabla_x f_{ik}^{a}\left(\mathbf{x}^{l}\right)^{T}(\mathbf{x}^{l}-\mathbf{x}^{f})-f_{ik}^{a}\left(\mathbf{x}^{l}\right)\right]y_{ik}&\\
\nabla_x h_{ik}^{a}\left(\mathbf{x}^{l}\right)^{T}\mathbf{x}^{a}\le\nabla_x h_{ik}^{a}\left(\mathbf{x}^{l}\right)^{T}\mathbf{x}^{f}+\left[\nabla_x h_{ik}^{a}\left(\mathbf{x}^{l}\right)^{T}(\mathbf{x}^{l}-\mathbf{x}^{f})-h_{ik}^{a}\left(\mathbf{x}^{l}\right)\right]y_{ik},l=1,\dots,L&(7)\\
\mathbf{x}=(\mathbf{x}^{g},\mathbf{x}^{s})\in\mathbb{R}^{n},\mathbf{y}\in\{0,1\}^{m}&\\
0\le\alpha^{g},\alpha_{ik}^{a},\qquad i\in D_k,k\in SD&\\
\mathbf{x}^{LO}\le\mathbf{x}^{g}\le\mathbf{x}^{UP}&
\end{array}
$$

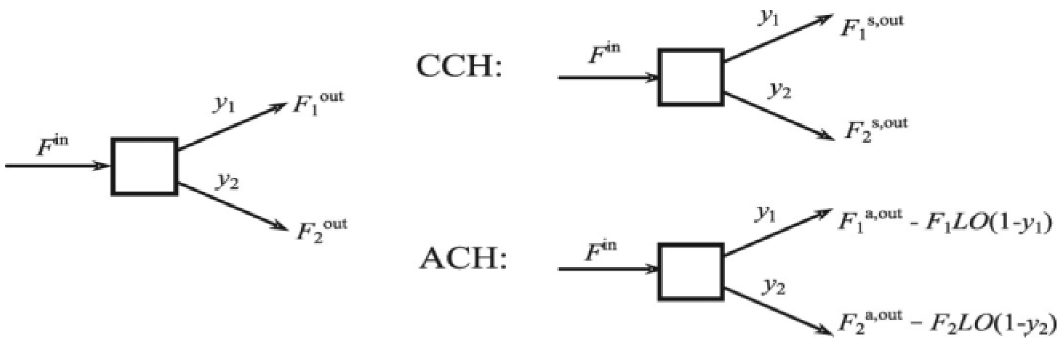

Figure 5. A single-choice splitter.

There are three possible ways of defining outer approximations (OAs) as inequalities (7) for the alternative formulation: (i) directly from (CCH-MILP) or (ii) directly from problem (A-CHRP) or (iii) with a combination of lineriza-tions from selected and rejected alternatives in problem (A-OAMP); proofs are provided in Appendix B. When alternatives are not selected, linearizations (7) are reduced to auxiliary linear inequalities $\left(\nabla_x f_{ik}^{a}(\mathbf{x}^l)^{T}\mathbf{x}^{a} \le \nabla_x f_{ik}^{a}(\mathbf{x}^l)^{T}\mathbf{x}^{f}\right)$ and $\left(\nabla_x h_{ik}^{a}(\mathbf{x}^l)^{T}\mathbf{x}^{a} \le \nabla_x h_{ik}^{a}(\mathbf{x}^l)^{T}\mathbf{x}^{f}\right)$ which preserve the feasibility of the OAs when $\mathbf{x}$ is set to $\mathbf{x}^{f}$. This enables the use of variables with nonzero lower bounds. Note that if $\mathbf{x}^{f}$ is set to $\mathbf{x}^{LO}$, mixed-integer inequalities (4) are reduced to simple lower bounds, and if it is set to $\mathbf{x}^{UP}$, inequalities (5) are reduced to the upper bounds. If the lower bounds and $\mathbf{x}^{f}$ are zero, problem (ACH-MILP) is reduced to problem (CCH-MILP). An interesting feature of the proposed formulation of OAs (inequality 7) is that nonzero $\mathbf{x}^{f}$ can be chosen in such a manner that linearization coefficients at y_{ik} become zero, and the mixed-integer OAs become pure-continuous constraints, which are potentially easier to solve, especially when the numbers of binary variables and linearizations are very large. It becomes obvious that the selection of $\mathbf{x}^{f}$ and, especially, the selection of the most suitable OAs and modeling representation may not be a straightforward task and may significantly influence the efficiency of the search.

Implementation of different modeling formulations in MIPSYN

Differences between the conventional and the alternative convex hull formulations are demonstrated by the simple example of a single-choice splitter. On the basis of this example, the general model for interconnection nodes and the MINLP model for process flowsheets were developed and implemented in the process synthesizer MIPSYN.

Example 2: Different Convex Hull Models for a Single-Choice Splitter. In Figure 5, which shows the flowchart for a single-choice splitter, either outflow F_1^{out} or F_2^{out} can be equal to F^{in}. The conventional convex hull formulation and the alternative convex hull representation with $\mathbf{x}^{f} = \mathbf{x}LO$ are given below:

Conventional convex hull:

$$F^{in} = F_1^{s,out} + F_2^{s,out} \tag{8}$$

$$FLO_1^{out} y_1 \le F_1^{s,out} \tag{9}$$

$$F_1^{s,out} \le FUP_1^{out} y_1 \tag{10}$$

$$FLO_2^{out} y_2 \le F_2^{s,out} \tag{11}$$

$$F_2^{s,out} \le FUP_2^{out} y_2 \tag{12}$$

$$y_1 + y_2 = 1 \tag{13}$$

$$\begin{aligned} &0 \le F^{in} \\ &F^{in} \le F^{in,UP} \end{aligned} \tag{14}$$

Alternative convex hull:

$$F^{in} = \left[F_1^{a,out} - FLO_1^{out}(1-y_1)\right] + \left[F_2^{a,out} - FLO_2^{out}(1-y_2)\right] \tag{15}$$

$$FLO_1^{out} + (FLO_1^{out} - FLO_1^{out})y_1 \le F_1^{a,out} \rightarrow F_1^{LO} \le F_1^{a,out} \tag{16}$$

$$F_1^{a,out} \le FLO_1^{out} + (FUP_1^{out} - FLO_1^{out})y_1 \tag{17}$$

$$FLO_2^{out} + (FLO_2^{out} - FLO_2^{out})y_2 \le F_2^{a,out} \rightarrow F_2^{LO} \le F_2^{a,out} \tag{18}$$

$$F_2^{a,out} \le FLO_2^{out} + (FUP_2^{out} - FLO_2^{out})y_2 \tag{19}$$

$$y_1 + y_2 = 1 \tag{20}$$

$$\begin{aligned} &F^{in,LO} \le F^{in} \\ &F^{in} \le F^{in,UP} \end{aligned} \tag{21}$$

where FLO_1^{out}, FLO_2^{out}, FUP_1^{out}, and FUP_2^{out} are nonzero scalars representing the outflow bounds. Note that in the alternative convex hull formulation, since $\mathbf{x}^{f} = \mathbf{x}^{LO}$, mixed-integer constraints 16 and 18 are reduced to simple lower bounds constraints. Consequently, the nonzero lower bounds can thus be applied directly even when an outflow is not selected. Similarly, in eqs. 17 and 19 can be replaced by upper bounds constraints if $\mathbf{x}^{f}$ is set to $\mathbf{x}^{UP}$. However, when $\mathbf{x}^{f}$ is set to an arbitrary value between the bounds in the alternative formulation, the number of mixed-integer bounding constraints remains the same as in the conventional formulation. Note that the additional terms like $x^{LO}(1 - y)$ must be introduced in the mass balance (15) to cancel out rejected flows; for example, if $F_1^{a,out}$ is selected ($y_1 = 1$, $y_2 = 0$), inequalities 18 and 19 give $F_2^{a,out} = F_2^{out.LO}$ and the last two terms in Eq. 15, representing the original outflow $F_2^{s,out}$, precisely cancel each other out, rendering $F^{in} = F_1^{a,out}$. It is interesting to note that if $F_1^{out,LO} = F_2^{out,LO}$, the mixed-integer balance Eq. 15 is reduced to the continuous one:

$$F^{in} = F_1^{out} + F_2^{out} - FLO_{1(2)}^{out} \quad (22)$$

Also, it is worth to note that when using ACH formulation a value of a local variable $\mathbf{x}^a$ has an exact physical meaning only when $y = 1$. However, when $y = 0$, $\mathbf{x}^a$ becomes equal to a nonzero value $\mathbf{x}^f$. To obtain a solution with exact physical meaning when solving alternative problem (ACH-MILP), it is recommended to translate the local variables $\mathbf{x}^a$ back to their original variables $\mathbf{x}^s$ using the same translating formula 1; for example in Figure 5, when ($y_1 = 1$, $y_2 = 0$) and $F_2^{a,out} = F_2^{out.LO}$, the physically meaningful variable $F_2^{s,out}$ is equal to zero ($F_2^{s,out} = F_2^{a,out} - F_2^{out.LO}\ (1 - y_2)$).

General Models for Interconnection Nodes. On the basis of the observation from the example of a single-choice splitter, general models for single-choice splitters with i outflows and single-choice mixers with i inflows were developed for $x^f = x^{LO}$:

$$x^{in} = \sum_{i \in D_k} \left(x_i^{a,out} - x_i^{out,LO}(1 - y_i) \right)$$
$$x_i^{a,out} \le x_i^{out,LO} + \left(x_i^{out,UP} - x_i^{out,LO} \right) y_i, \quad \forall i \in D_k \quad \text{(splitter)}$$
$$\sum_{i \in D_k} y_i = 1 \quad \left(x^{in,LO}, x_i^{out,LO} \right) \le \left(x^{in}, x_i^{a,out} \right) \le \left(x^{in,UP}, x_i^{out,UP} \right)$$
$$x^{out} = \sum_{i \in D_i} \left(x_i^{a,in} - x_i^{in,LO}(1 - y_i) \right)$$
$$x_i^{a,in} \le x_i^{in,LO} + \left(x_i^{in,UP} - x_i^{in,LO} \right) y_i, \ \forall i \in D_k \quad \text{(mixer)}$$
$$\sum_{i \in D_k} y_i = 1 \quad \left(x^{out,LO}, x_i^{in,LO} \right) \le \left(x^{out}, x_i^{a,in} \right) \le \left(x^{out,UP}, x_i^{in,UP} \right)$$

where x can represent process parameters like flows, component flows, temperature, pressure and/or vapor pressure.

General MINLP Models for Process Superstructures. Regarding process superstructures consisting of interconnection nodes $k \in SD$ and alternative process unit nodes $i \in D_k$, the following MINLP problem (PF-MINLP) and its translated form (APF-MINLP) at $\mathbf{x}^f = \mathbf{x}^{LO}$ for process superstructures are obtained:

$$\begin{array}{l}
\text{(PF-MINLP):} \\
\min Z = \sum_{i \in D_k} \sum_{k \in SD} \left(c_{ik} y_{ik} + f_{ik}^a(\mathbf{x}^s) \right) + f^g(\mathbf{x}^g) \\
\text{s.t. } \mathbf{h}^g(\mathbf{x}^g) \le \mathbf{0} \\
\quad \mathbf{A}^g(\mathbf{x}^g) \le \mathbf{b}^g \\
\quad \mathbf{E}^g(\mathbf{y}) \le \mathbf{e}^g \\
\quad \mathbf{A}^r(\mathbf{x}^g, \mathbf{x}^s) \le \mathbf{b}^r \\
\quad \left. \begin{array}{l} xLOy_{ik} \le \mathbf{x}^s \\ x^s \le xUPy_{ik} \end{array} \right\} \forall \mathbf{x}^s \in \mathbf{X}_{ik} \\
\quad A^{ik}(\mathbf{x}^s) \le b_{ik}\, y_{ik} \\
\quad h_{ik}(\mathbf{x}^s) \le 0 \qquad i \in D_k, K \in SD \\
\quad \mathbf{x}^g, \mathbf{x}^s \in \mathbb{R}^n, \mathbf{y} \in \{0,1\}^m \\
\quad \mathbf{x}^{LO} \le \mathbf{x}^g \le \mathbf{x}^{UP}
\end{array}
\xrightarrow{x^s = x^a - x^f(1-y)}
\begin{array}{l}
\text{(APF-MINLP):} \\
\min Z = \sum_{i \in D_k} \sum_{k \in SD} \left(c_{ik} y_{ik} + f_{ik}^a\left(\mathbf{x}^a - \mathbf{x}^f(1 - y_{ik})\right) \right) + f^g(\mathbf{x}^g) \\
\text{s.t. } \mathbf{h}^g(\mathbf{x}^g) \le \mathbf{0} \\
\quad \mathbf{A}^g(\mathbf{x}^g) \le \mathbf{b}^g \\
\quad \mathbf{E}^g(\mathbf{y}) \le \mathbf{e}^g \\
\quad \mathbf{A}^r(\mathbf{x}^g, \Phi_{x^s}(\mathbf{x}^a, \mathbf{y})) \le \mathbf{b}^r \\
\quad \left. \begin{array}{l} \mathbf{x}^{a,LO} \le x^a \\ \mathbf{x}^a \le \mathbf{x}^{LO} + (xUP - \mathbf{x}^f) y_{ik} \end{array} \right\} \forall \mathbf{x}^s \in \mathbf{X}_{ik} \\
\quad A^{ik}(\mathbf{x}^a - \mathbf{x}^f(1 - y_{ik})) \le b_{ik}\, y_{ik} \\
\quad h_{ik}(\mathbf{x}^a - \mathbf{x}^f(1 - y_{ik})) \le 0 \qquad i \in D_k, K \in SD \\
\quad \mathbf{x}^g, \mathbf{x}^a \in \mathbb{R}^n, \mathbf{y} \in \{0,1\}^m \\
\quad \mathbf{x}^{LO} \le \mathbf{x}^g \le \mathbf{x}^{UP}
\end{array}$$

As can be seen, in problem (APF-MINLP) alternative variables with nonzero lower bounds can now be used in contrast to the problem (PF-MINLP).

Implementation in Process Synthesizer MIPSYN. Until recently only the big-M formulation of OAs and the big-M representation of logical interconnection nodes (single-choice mixers and splitters) were used in the MINLP process synthesizer MIPSYN to solve MINLP synthesis problems. At present, the conventional convex hull and the alternative convex hull formulations are implemented in MIPSYN, too. Two additional libraries of models for process unit and interconnection nodes were developed and two preprocessors were programmed to derive different convex hull OAs. The models are formulated in the most generalized form using various capabilities of the high-level language of GAMS. In this way data-and-topology independent models were developed.

Example 3: Illustrative Flowsheet Problem. To demonstrate different model formulations and the idea of the translation of variables, a simple flowsheet problem by Kocis and Grossmann[27] is used. Figure 6 shows a simple superstructure comprising two reactors and only one can be selected. The objective is to minimize the total cost at a fixed demand of the final outflow. The original MINLP model is given below:

$$\begin{aligned}
&\min Z = 7.5y_1 + 5.5y_2 + 7V_1 + 6V_2 + 5x \\
&\text{s.t. } z_1 = 0.9(1 - e^{-0.5V_1}) \cdot x_1 \\
&\quad z_2 = 0.8(1 - e^{-0.4V_2}) \cdot x_2 \\
&\quad z_1 + z_2 = 10 \\
&\quad x_1 + x_2 = x \\
&\quad V_1 \le 10y_1 \quad V_2 \le 10y_2 \qquad \text{(MINLP)} \\
&\quad x_1 \le 20y_1 \quad x_2 \le 20y_2 \\
&\quad y_1 + y_2 = 1 \\
&\quad V_1, x_1, z_1, V_2, x_2, z_2 \ge 0 \\
&\quad y_1, y_1 \in \{0, 1\}
\end{aligned}$$

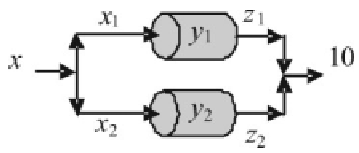

Figure 6. Superstructure of the illustrative example.

This model was solved by using the OA/ER algorithm and the modeling/decomposition (M/D) strategy which is a special form of the GDP and useful for process scheme synthesis. Applying mathematical forms of the conventional and alternative convex hull MILP master problems to the original model, the following formulations were obtained:

$$\begin{aligned}
&(\text{CCH-MILP}):\\
&\min z = 7.5y_1 + 5.5y_2 + 7V_1^{s} + 6V_2^{s} + 5x\\
&\text{s.t.}\\
&z_1^{s} \le 0.781x_1^{s} + 0.888V_1^{s} - 3.693y_1\\
&z_2^{s} \le 0.672x_2^{s} + 0.761V_2^{s} - 3.584y_2\\
&z_1^{s} + z_2^{s} = 10\\
&x_1^{s} + x_1^{s} = x\\
&VLO_1 y_1 \le V_1^{s}\\
&V_1^{s} \le VUP_1 y_1\\
&VLO_2 y_2 \le V_2^{s}\\
&V_2^{s} \le VUP_2 y_2\\
&xLO_1 y_1 \le x_1^{s}\\
&x_1^{s} \le xUP_1 y_1\\
&xLO_2 y_2 \le x_2^{s}\\
&x_2^{s} \le xUP_2 y_2\\
&zLO_1 y_1 \le z_1^{s}\\
&z_1^{s} \le zUP_1 y_1\\
&zLO_2 y_2 \le z_2^{s}\\
&z_2^{s} \ge zUP_2 y_2\\
&y_1 + y_2 = 1
\end{aligned}$$

$$\xrightarrow[x^{f} = x^{LO}]{x^{s} = x^{a} - x^{f}(1-y)}$$

$$\begin{aligned}
&(\text{ACH-MILP}):\\
&\min z = 7.5y_1 + 5.5y_2 + 7V_1^{a} - 7V_1^{LO}(1-y_1)\\
&\qquad + 6V_1^{a} - 6V_2^{LO}(1-y_2) + 5x\\
&\text{s.t.}\\
&z_1^{a} \le 0.781x_1^{a} + 0.888V_1^{a} - 0.342 - 2.949y_1\\
&z_2^{a} \le 0.672x_2^{a} + 0.761V_2^{a} + 0.016 - 3.650y_2\\
&\left(z_1^{a} - z_1^{LO}(1-y_1)\right) + \left(z_2^{a} - z_2^{LO}(1-y_2)\right) = 10\\
&\left(x_1^{a} - x_1^{LO}(1-y_1)\right) + \left(x_2^{a} - x_2^{LO}(1-y_2)\right) = x\\
&V_1^{a} \le V_1^{LO} + (VUP_1 - V_1^{LO})y_1\\
&V_2^{a} \le V_2^{LO} + (VUP_2 - V_2^{LO})y_2\\
&x_1^{a} \le x_1^{LO} + (xUP_1 - x_1^{LO})y_1\\
&x_2^{a} \le x_2^{LO} + (xUP_2 - x_2^{LO})y_2\\
&z_1^{a} \le z_1^{LO} + (zUP_1 - z_1^{LO})y_1\\
&z_2^{a} \le z_2^{LO} + (zUP_2 - z_2^{LO})y_2\\
&y_1 + y_2 = 1\\
&V_1^{LO}, V_2^{LO}, x_1^{LO}, x_2^{LO}, z_1^{LO}, z_2^{LO} \le V_1^{a}, V_2^{a}, x_1^{a}, x_2^{a}, z_1^{a}, z_2^{a}
\end{aligned}$$

Note again that in the alternative problem (ACH-MILP), unlike the conventional convex hull modeling representation (CCH-MILP), nonzero lower bounds can be used directly. By setting $\mathbf{x}^{f} = \mathbf{x}^{LO}$, one-half of the mixed-integer bounding constraints are replaced by simple bounds. However, additional terms like $x^{LO}(1 - y)$ must be introduced to satisfy feasibility of constraints. In this way, the mass balances become mixed-integer constraints to cancel out rejected flows. However, mixed-integer constraints can be reduced to continuous ones by setting lower bounds to the same values. The corresponding big-M model and the entire solution procedure for all three model formulations are given in Appendix C.

Examples

Three synthesis problems of different sizes and complexities have been solved to test and compare the efficiencies of all three OAs and modeling representations. To answer the questions which representation is the most efficient and why, various experiments were performed on all three examples.

The first example is a network synthesis problem with a simple model but very large-scale combinatorics with 400 binary variables. The second example is the synthesis of a heat exchanger network (HEN) comprising different types of exchanger. The model exhibits moderate complexity and high combinatorics (249 and 371 binary variables). The last, namely the allyl chloride example, is the synthesis of a reactor/separator network in an overall heat integrated process scheme, with a complex model and moderate-size combinatorics [172 and 184 binary variables, when all the final elements in the plug-flow reactors (PFR) are included].

Network synthesis problem

This numerical problem is an extension of the small illustrative flowsheet problem, namely Example 3.[27] Additional pairs of reactors were added to the superstructure. The superstructure thus comprised 200 pairs of interconnected reactors (400 binary variables) and in addition, operational costs were added to the objective function (Eq. 23). Figure 7 shows a superstructure comprising a sequence of exclusive-or reactor alternatives. The objective was to minimize the total cost at a fixed demand of the final outflow; $\mathbf{x}^{f}$ was set to $\mathbf{x}^{LO}$.

$$\min z = c_{fi}^{1} y_i^{1} + c_{v}^{1} V_i^{1} + c_{fi}^{2} y_i^{2} + c_{vi}^{2} V_i^{2} + c_{oi}^{1} x_i^{1} + c_{oi}^{2} x_i^{2} + 5x_1 \quad i \in 200 \qquad (23)$$

The solution statistics up to the third major MINLP iteration is reported in Table 1. As can be seen in Table 1, the big-M formulation cannot solve the problem in a reasonably short time, whilst both convex hull representations enable the solving of this highly combinatorial problem very

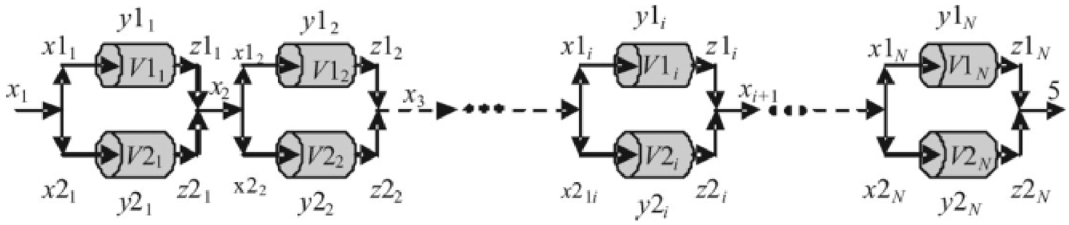

Figure 7. Superstructure of the network synthesis problem.

quickly. Note that with the same integrality gap and a smaller number of rows, the alternative formulation could solve the problem in only one third of the CPU time needed to solve MILP master problems using the conventional convex hull formulation. In addition, the CPU time for solving NLP subproblems in the ACH formulation is a quarter of the CPU time needed to solve the problem using the CCH formulation. It should be noted that the number of rows and variables for NLP subproblems in the tables represents single equations and variables as shown in GAMS model statistics, while in MILP it represents the rows and columns of the reduced MILP master problem obtained after substitution of variables and the deleting the redundant rows by a MILP solver preprocessor.

HEN synthesis problem

When the HEN synthesis is performed with alternative exchanger types, each match in a stage-wise superstructure (Figure 8) is comprised of a double pipe, a plate and frame, a shell and tube exchanger, and a by-pass. The model is described in detail by Sorsak and Kravanja.[28]

Consideration of different types of exchanger enables the simultaneous selection of exchanger types; however, it significantly increases the number of binary variables. In an example of four hot (index *i*) and five cold (index *j*) process streams, four stages (index *k*) and four match alternatives (index *l*), the problem originally had 320 *y*s. By prescreening the alternatives, the number was reduced to 249. Disaggregated variables are match heat exchanger inlet and outlet temperatures of hot (Ti^{A}_{ijkl}, Ti^{B}_{ijkl}) and cold streams (Tj^{A}_{ijkl}, Tj^{B}_{ijkl}), match temperature differences for heat exchangers (ΔT_{ijkl}), match heat exchanger heat loads (Ql_{ijkl}), and heat exchanger costs (cv_{ijkl}).

Table 2 shows the statistics (a) for the aforementioned four hot–five cold stream problem and (b) for a problem with six cold and five hot streams in four stages containing 371 binary variables. In both cases $\mathbf{x}^{f}$ was set to $\mathbf{x}^{LO}$. With respect to the integrality gap, number of iterations, CPU time and number of nodes for solving MILP master problems, both convex hull representations in case (a) significantly outperform the big-M one, whilst the efficiency of the alternative convex hull formulation is approximately twice the one of the conventional formulation. It is also interesting to note that when solving NLP subproblems the ACH formulation outperforms the Big-M and CCH formulations too: the CPU time and number of iterations of the ACH formulation are about one half of the other two formulations.

When a somewhat larger problem (b) was considered, the CPU time of solving the MILP master problems with the alternative formulation was less than half an hour, whilst performing the same number of main MINLP iterations and solving the problem with the conventional formulation only to the sub-optimal solution took more than an hour of the CPU time. In this case, the CPU time required to solve NLP subproblems for ACH formulation was one quarter of the CPU time needed to solve the problem using the CCH formulation. It is interesting to note that with the Big-M formulation the problem could not be solved in a reasonably short time.

Allyl chloride example

Details of the allyl chloride problem are given by Bedenik et al.[29] The reactor/separator superstructure (Figure 9) comprises a sequence of PFR/CSTRs with side streams and intermediate separators at different locations. Each PFR consists of a train of several alternative elements. The corresponding DAE system is modeled by the orthogonal collocation on finite elements. Simultaneous heat integration was performed by the use of the HEN synthesis model, proposed by Yee and Grossmann.[30] The objective was to maximize the net present value at a fixed production of allyl chloride. The overall model is highly nonlinear and nonconvex. Therefore, many numerical and other issues are present which makes the comparison between formulations more difficult, e.g., due to the effects of nonconvexities it is impossible to compare different formulations based on an integrality gap.

The problem was first solved (a) for one final element per PFR (172 binary variables) and then (b) with all final

Table 1. Solution Statistics of the Reactor Network Synthesis Problem (400 *y*s)

			No. of Rows/Var.		No. of Iterations for 3 Iterations		No. of Nodes for 3 Iterations	Nodes/s for 3 Iterations	CPU for 3 Iterations (s)	
	Best NLP	Int. Gap (%)	NLP	MILP	NLP	MILP	MILP	MILP	NLP	MILP
Big-M	n/a	n/a	1802/2003	4393/1598	*77	n/a	n/a	n/a	*0.7	n/a
CCH	183.87	0.868	3001/1401	2797/1199	750	21878	292	19.3	3.2	15.1
ACH	183.87	0.868	1201/1401	1798/1398	58	4417	293	62.3	0.7	4.7

CPLEX/GAMS version 21.7, processor PENTIUM 4 2.81 GHz, 512 MB of RAM.
*Only the first NLP.

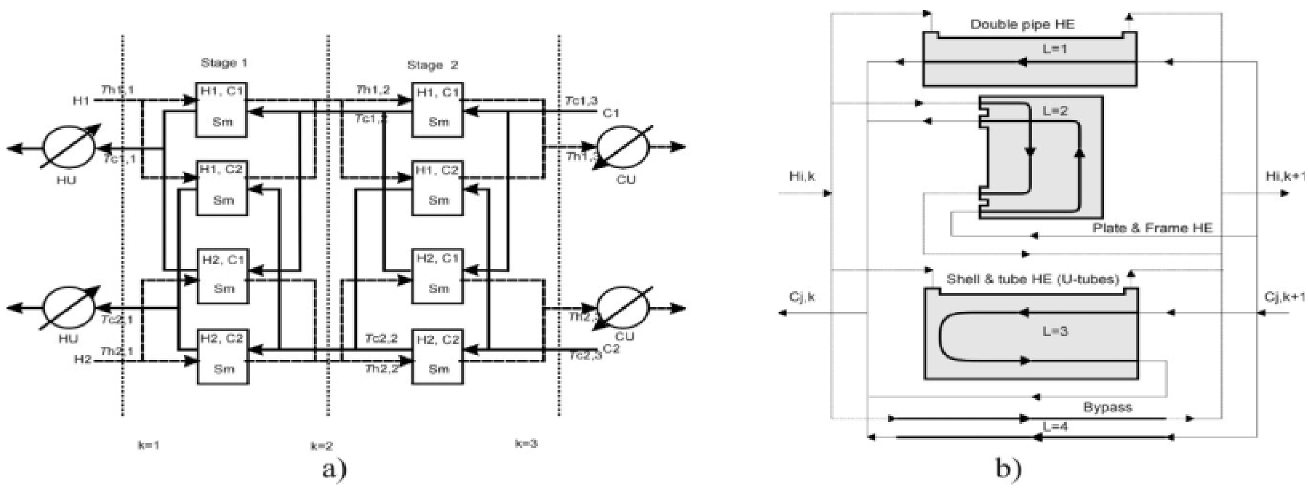

Figure 8. Heat exchanger network.

(a) Superstructure of HEN. (b) Match superstructure.

elements (184 binary variables). Table 3 shows the solution statistics up to the 17th major MINLP iteration for both cases; $\mathbf{x}^{\mathrm{f}}$ was set to $\mathbf{x}^{\mathrm{LO}}$. As can be seen in Table 3(a) for the MILP steps both convex hull formulations are very similar in performance and have twice the efficiency of the Big-M formulation for the MILP step. However, when all final elements are applied [Table 3(b)], the efficiency in solving the MILP master problems with the ACH formulation is almost twice as good as the one of with CCH and four times better than in the Big-M. With respect to the solution of NLP subproblems, the efficiency of the ACH formulation is again approximately twice as good as the one of CCH. It should be noted that the selection of the optimal final element in the PFR is formulated by big-M constraints so that the overall process ACH and CCH formulations are in fact combined ACH/Big-M and CCH/Big-M formulations.

Improving the Efficiency of the Alternative Formulation

The selection of $\mathbf{x}^{\mathrm{f}}$ is not a straightforward task and it may significantly affect the efficiency of the alternative MILP master formulation. On the other hand, when we are dealing with strictly local variables the translation of variables may not be applied completely, which may significantly affect efficiency. Therefore, some additional experiments were performed to improve further the efficiency of the alternative MILP master problems.

Selection of x^f

To study the efficiency of the ACH formulation with respect to the selection of $\mathbf{x}^{\mathrm{f}}$, different experiments were carried out with $\mathbf{x}^{\mathrm{f}}$ defined anywhere within the bounds:

$$\mathbf{x}^{\mathrm{f}} = \mathbf{x}^{\mathrm{LO}} + \delta(\mathbf{x}^{\mathrm{UP}} - \mathbf{x}^{\mathrm{LO}}) \tag{24}$$

where δ is a scalar in the interval from 0 to 1. When $\delta = 0$, $\mathbf{x}^{\mathrm{f}}$ is at the lower bound, as in all our former cases, when $\delta = 1$, $\mathbf{x}^{\mathrm{f}}$ is at the upper bound, and when $0 < \delta < 1$, $\mathbf{x}^{\mathrm{f}}$ lies between the bounds.

The above equation was applied to the network synthesis example and the HEN example, in the latter case with somewhat different data. Solution statistics for both examples are reported in Table 4. As can be seen, the selection of $\mathbf{x}^{\mathrm{f}}$ is very important for the efficiency of the ACH. In the network synthesis example [Table 4(a)], the best efficiency of the search is achieved when $\mathbf{x}^{\mathrm{f}} = \mathbf{x}^{\mathrm{LO}}$. When $\mathbf{x}^{\mathrm{f}} = \mathbf{x}^{\mathrm{UP}}$, the CPU time is comparable with the one of the conventional convex

Table 2. Solution Statistics for Two HEN Synthesis Problems

			No. of Rows/Var.		No. of Iterations for 15 Iterations		No. of Nodes for 15 Iterations	Nodes/s for 15 Iterations	CPU for 15 Iterations (s)	
	Best NLP	Int. Gap (%)	NLP	MILP	NLP	MILP	MILP	MILP	NLP	MILP
(a) 249 binary variables (5 cold streams, 4 hot streams, 4 stages)										
Big-M	884.07	1.548	8342/4232	2115/1089	234	2248581	70257	140.3	12.7	500.9
CCH	818.69	0.607	6342/4232	1613/572	224	612529	35150	215.2	12.9	163.3
ACH	818.69	0.607	4182/4232	1317/884	168	321062	19411	231.6	5.7	83.8
(b) 371 binary variables (6 cold streams, 5 hot streams, 4 stages)										
Big-M	n/a	n/a	8345/4235	5166/1961	*6	n/a	n/a	n/a	*0.2	n/a
CCH	1925.57	68.726	9458/6318	3668/1390	957	6301759	415404	109.4	23.7	3795.4
ACH	1923.01	68.726	6218/6318	3040/2170	131	2778973	153242	103.6	8.0	1478.9

CPLEX/GAMS version 21.7, processor PENTIUM 4 2.81 GHz, 512 MB of RAM
*Only the first NLP.

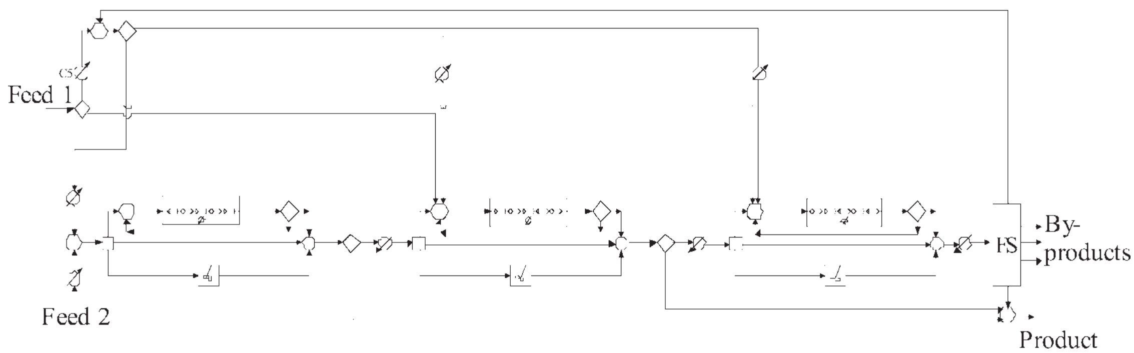

Figure 9. Reactor/separator superstructure of the allyl chloride problem.

hull formulation although it requires only one-half of the iterations needed to solve the problem by the CCH. As soon as $\mathbf{x}^{\mathrm{f}}$ slightly differs from $\mathbf{x}^{\mathrm{LO}}$ (refer ACH at $\delta = 10^{-6}$ in Table 4), the efficiency of the search is significantly decreased since the number of rows in the reduced MILP problem is increased and more pivot operations in the LPs at the nodes are then required. In the HEN example, on the other hand, the ACH formulation shows the best performance when $\mathbf{x}^{\mathrm{f}}$ is exactly in the middle between $\mathbf{x}^{\mathrm{LO}}$ and $\mathbf{x}^{\mathrm{UP}}$, which perhaps indicates that some variables should have $\mathbf{x}^{\mathrm{f}} = \mathbf{x}^{\mathrm{LO}}$ and others $\mathbf{x}^{\mathrm{f}} = \mathbf{x}^{\mathrm{UP}}$. It should be noted that different hypotheses were postulated to select the most suitable $\mathbf{x}^{\mathrm{f}}$. However, a great deal of experiments could not confirm these hypotheses. It seems that the relation $\mathbf{x}^{\mathrm{f}} = \mathbf{x}^{\mathrm{LO}}$ can thus be regarded as the only suitable heuristic rule proposed so far.

Further reduction of the MILP master problem

The translation of variables and hence the entire local part of the problem (ACH-MILP) may not be strictly applied to all variables of alternatives but only to a subset of variables through which alternatives communicate with the global part of the model, i.e., disaggregated variables appearing in the interconnection nodes and variables appearing in objective function terms. Indeed, for strictly local variables which appear only within modules of alternatives, the relation $\mathbf{x}^{\mathrm{a}} = \mathbf{x}^{\mathrm{f}}$ for cases when alternatives are rejected, is not compulsory. In addition, the translation of variables can be performed only partially, i.e., $\mathbf{x}^{\mathrm{s}} = \mathbf{x}^{\mathrm{a}}$ without the use of the second term $\mathbf{x}^{\mathrm{f}}(1 - y)$, since strictly local variables do not affect the global part of the model and the objective function. Consequently, their bounding logical constraints as in Eqs. 4 and 5 may be neglected, which could lead to a significant reduction in the number of rows in the reduced MILP master problem when the number of strictly local variables is high. It should be noted that since in the reactor network and HEN synthesis problems all local variables appear also in the interconnection node constraints and objective function, the entire problem (ACH-MILP) was applied to them. However, since most of the local variables in the case of the allyl chloride example appear only in modules of alternatives, only the necessary part of the problem (ACH-MILP) was applied to it. To demonstrate the efficiency of the reduced problem one should consider the reactor network example, where the reactor model is extended by the following heat flow and conversion equations:

$$\Phi 1_i^a = -\Delta_r H \cdot z1_i^a \text{ and } \Phi 2_i^a = -\Delta_r H \cdot z2_i^a, \tag{25}$$

$$X1_i^a = \frac{z1_i^a}{x1_i^a} \text{ and } X2_i^a = \frac{z2_i^a}{x2_i^a}. \tag{26}$$

where heat flow rates Φ_i^{a} and conversions X_i^{a} appear only locally. Table 5 shows the solution statistics until the third major MINLP iteration. As can be seen, the difference in the efficiency of the search is an order of magnitude in favor of the reduced ACH formulation. Another interesting feature is about the handling of nonconvexities when NLPs are applied to the whole superstructure, i.e., even to the rejected units. If the translation of strictly local variables $\mathbf{x}^{\mathrm{s}}$ is applied only partially so that nonzero-lower-bounded variables $\mathbf{x}^{\mathrm{a}}$ are not accompanied by the terms $\mathbf{x}^{\mathrm{f}}(1-y)$, then the problem of mathematical singularities when some units are rejected and some original

Table 3. Solution Statistics of the Allyl Chloride Problem

			No. of Rows/Var.		No. of Iterations for 17 Iterations		No. of Nodes for 17 Iterations	Nodes/s for 17 Iterations	CPU for 17 Iterations (s)	
	Best NLP	Int. Gap (%)	NLP	MILP	NLP	MILP	MILP	MILP	NLP	MILP
(a) 172 binary variables (Yee + 1 finite element)										
Big-M	81.983	0.929	1513/3600	2167/2003	1874	401552	2528	12.5	16.1	202.8
CCH	82.193	2.869	1573/3357	2200/1980	2432	170706	2201	19.6	18.9	112.5
ACH	82.830	0.905	1659/3704	1109/1130	2328	121612	3872	34.9	17.0	111.0
(b) 184 binary variables (Yee + selection of optimal finite element)										
Big-M	81.924	0.190	1040/3098	2307/2193	2725	4525051	19830	9.2	25.0	2166.5
CCH/Big-M	81.836	100.00	1283/3127	2363/2183	3911	1503644	8053	8.4	41.2	960.3
ACH/Big-M	81.769	0.343	1648/3697	1230/1313	2996	854049	7700	14.6	24.1	529.2

CPLEX/GAMS version 21.7, processor PENTIUM 4 2.81 GHz, 512 MB of RAM.

Table 4. Solution Statistics for the Selection of Different $\mathbf{x}^f$

	Best NLP	Int. Gap (%)	No. of Rows/Var. NLP	No. of Rows/Var. MILP	No. of Iterations NLP	No. of Iterations MILP	No. of Nodes MILP	Nodes/s MILP	CPU (s) NLP	CPU (s) MILP
(a) Reactor network synthesis problem (three major MILP iterations)										
Big-M	n/a	n/a	1802/2003	4393/1598	*77	n/a	n/a	n/a	*0.7	n/a
CCH	183.870	0.868	3001/1401	2797/1199	750	21878	292	19.3	3.2	15.1
ACH ($\delta = 0$), $\mathbf{x}^{LO}$	183.870	0.868	1201/1401	1798/1398	58	4417	293	62.3	0.7	4.7
ACH ($\delta = 1 \times 10^{-6}$)	183.870	0.868	1201/1401	2797/1199	58	9806	299	29.8	0.7	10.0
ACH ($\delta = 0.1$)	183.870	0.868	1201/1401	2797/1199	158	13902	374	30.7	1.4	12.2
ACH ($\delta = 0.5$)	183.870	0.868	1201/1401	2797/1199	150	87977	2020	45.2	1.4	44.7
ACH ($\delta = 0.9$)	183.870	0.868	1201/1401	2797/1199	133	66773	1637	39.4	1.4	41.5
ACH ($\delta = 1.0$), x^{UP}	183.870	0.869	1201/1401	1797/1399	145	8993	292	16.0	1.4	18.2
(b) HEN synthesis problem (21 major MILP iterations)										
Big-M	n/a	n/a	n/a	3414/1314	*6	n/a	n/a	n/a	*0.2	n/a
CCH	778.89	78.121	6343/4233	2546/978	148	560967	42024	165.6	15.1	253.8
ACH ($\delta = 0$), $\mathbf{x}^{LO}$	779.74	78.121	4183/4233	2124/1524	107	542891	45769	196.3	6.0	233.2
ACH ($\delta = 0.1$)	784.67	78.121	6423/4233	2950/1118	146	454777	28430	123.7	6.2	229.9
ACH ($\delta = 0.5$)	783.53	78.121	6423/4233	2950/1118	147	331552	18898	102.7	6.5	184.0
ACH ($\delta = 0.9$)	782.29	78.121	6423/4233	2950/1118	100	425164	24031	125.9	6.2	190.9
ACH ($\delta = 1.0$), x^{UP}	783.10	78.121	6423/4233	2166/1566	106	431391	25427	146.3	9.0	173.8

CPLEX/GAMS version 21.7, processor PENTIUM 4 2.81 GHz, 512 MB of RAM.
*Only the first NLP.

variables $\mathbf{x}^s$ go to zero, is entirely overcome, since now $\mathbf{x}^a$ are forced to have lower bounds greater than zero. For example in Eq. 26 $X1_i^a$ and $X2_i^a$ have nonzero lower bounds and the singularities cannot occur. Note, however, that this incomplete translation of variables can be applied only to strictly local variables.

Conclusion

An alternative convex hull representation was developed which allows the direct definition of nonzero lower bounds ($\mathbf{x}^{LO} \leq \mathbf{x}^a \leq \mathbf{x}^{UP}$) for local variables and, hence, enables the solving of large-scale combinatorial problems in narrowed lifted space of variables. On the basis of the conventional and alternative formulations an advanced and robust synthesizer shell MIPSYN, capable of solving large-scale applications in different engineering domains, was developed. The earliest experiences with the alternative convex hull representation indicate that the alternative convex hull representation is in most cases more efficient with respect to the NLP and MILP steps (Figure 10) in solving high combinatorial problems than the conventional one. It has the smallest number of rows in reduced MILP master problems and in most cases the smallest number of equations in NLP steps. The ACH representation is particularly useful for solving problems with nonzero lower bounds where cones of local variables $\mathbf{x}^a$ are narrowed within the nonzero bounds. In cases when the lower bounds are zero and $\mathbf{x}^f = \mathbf{x}^{LO}$, the ACH representation is reduced to the conventional one. Figure 2 and Appendix A show that the convex hulls of the CCH and ACH representations are the same, and hence the tightness, which is confirmed also by the results in Tables 1 and 2. However, there are some other important attributes that affect the efficiency of the search, some more than others:

i) When binary variables in the MILP master problem tend to be nonbasis variables, this results in better efficiency. Indeed, in the case of the reactor network problem most of the binary variables in the ACH formulation were nonbasic variables. It was noted by examining the detailed display of the MILP phase that the CCH formulation makes an order of magnitude more calculations than the ACH one.

ii) The selection of $\mathbf{x}^f$ is very important. An inappropriate selection could severely decrease the efficiency of the MINLP search. Probably the best and the most obvious choice is $\mathbf{x}^f = \mathbf{x}^{LO}$, especially when problems are minimized since lower bounds are the naturally expected position of variables when alternatives are rejected.

iii) There is no doubt that the number of rows and variables, and thus, the sizes of the reduced MILP master problems, also have an important impact on efficiency. As can

Table 5. Solution Statistics of the Extended Reactor Network Problem with Additional Local Variables (400 *ys*)

	Best NLP	Int. Gap (%)	No. of Rows/Var. NLP	No. of Rows/Var. MILP	No. of Iterations for 3 Iterations NLP	No. of Iterations for 3 Iterations MILP	No. of Nodes MILP	Nodes/s for 3 Iterations MILP	CPU for 3 Iterations (s) NLP	CPU for 3 Iterations (s) MILP
Big-M	n/a	n/a	2602/2803	2398/1598	*30	n/a	n/a	n/a	*0.4	n/a
CCH	85.711	2.41	3801/2201	3595/1199	808	659632	33605	84.6	4.2	397.4
ACH ($\mathbf{x}^f = \mathbf{x}^{LO}$)	85.711	2.41	1835/2035	1798/1398	60	42801	5089	194.3	0.8	26.2

CPLEX/GAMS version 21.7, processor PENTIUM 4 2.81 GHz, 512 MB of RAM.
*Only the first NLP.

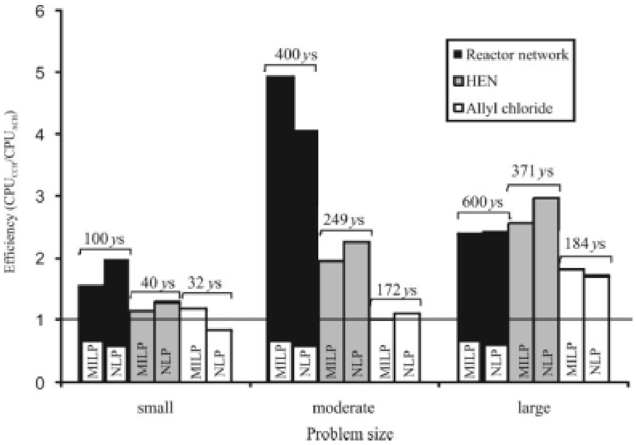

Figure 10. Efficiency in solving MILP master problems and NLP subproblems vs. problem size.

be seen from the example problems, the number of rows is always the smallest in ACH formulations when $\mathbf{x}^{\mathrm{f}} = \mathbf{x}^{\mathrm{LO}}$, especially when the bounding logical constraints in the problem (ACH-MILP) are applied only to a subset of variables appearing either as disaggregated variables or alternative's variables appearing in objective function terms. Additionally, in most cases in the ACH formulation the number of equations in the NLP subproblems is the smallest, too. Although the transformation of variables does not decrease a problem size, it may significantly decrease the number of rows and iterations in both the MILP master problems and NLP subproblems, thus contributing to the more efficient solution of LPs at the nodes.

iv) An increase in the lower bounds of the alternative's variables may improve the performance of ACH formulations over CCH ones. Increasing the lower bounds in the case of the reactor network problem dramatically improves the efficiency of the ACH vs. the CCH formulation (Figure 11).

Despite the mentioned advantages, ACH formulations exhibit the strongest sensitivity to the effects of nonconvexities and the model representations are the most complicated. The performances of different OAs and modeling representations are summarized in Table 6. The best choice for small size

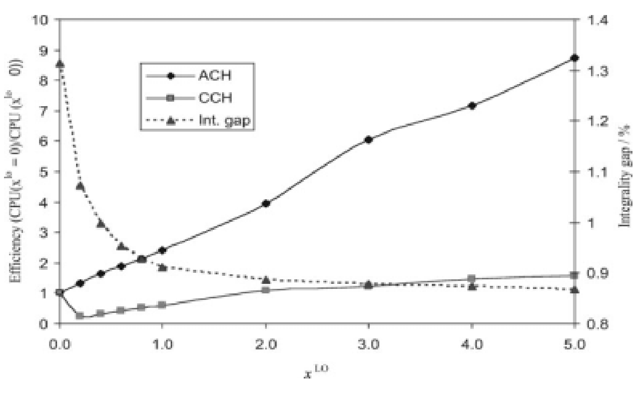

Figure 11. Efficiencies in solving MILP master problems of the ACH and CCH representations vs. the increase of the lower bounds.

Table 6. Performance of Different OAs and Modeling Representations

	Big-M	Convex hull	Alternative $\mathbf{x}^{\mathrm{f}} = \mathbf{x}^{\mathrm{LO}}$
Easiness of modeling	The easiest	Moderate	The most complicated
Problem size	From the smallest to moderate	The largest	The smallest or moderate
Effect of nonconvexities	The smallest	Moderate	The strongest
Nodes/s of CPU time	The largest	Moderate	The smallest or moderate
The best overall performance	For small problems	For moderate problems	For large problems

problems is believed to be the conventional convex hull representation because the model formulations are simpler than the alternative ones. On the other hand, when models are generated from a library of modules, as in the case with the equation-oriented modular synthesizer MIPSYN, the best choice is the ACH representation since it increases the likelihood of achieving the best efficiency of the MINLP search.

Acknowledgments

The authors are grateful to the Slovenian Ministry of Higher Education, Science and Technology, for their financial support (PhD research fellowship contract No. 3211-05-000566) and Funding of program group contract No. P2-0032.

Notation

A = linear constraints
B = matrix of coefficients in linear constraints
b = coefficients in linear constraints
c = cost coefficient (k$)
cv_{ijk} = variable charge coefficient for heat exchanger [$/(m^2 a)]
D_k = set of terms in disjunction k
E = logical constraints
e = coefficients in logical constraints
f = function
F = flow rate (mol/s)
FLO = nonzero scalar (mol/s)
g_{ik} = nonlinear constraints in convex hull
$h(\boldsymbol{x})$ = nonequality nonlinear constraint function
L = set of solutions
M = Big-M scalar
Ql_{ijkl} = match heat exchanger heat load (kW)
SD = set of disjunctions
T = Temperature (K)
Ti^{A}_{ijkl} = match heat exchanger inlet temperatures of hot stream (K)
Ti^{B}_{ijkl} = match heat exchanger outlet temperatures of hot stream (K)
Tj^{A}_{ijkl} = match heat exchanger inlet temperatures of cold stream (K)
Tj^{B}_{ijkl} = match heat exchanger outlet temperatures of cold stream (K)
V = volume (m^3)
VLO = nonzero scalar (m^3)
$\mathbf{x}$ = vector of variables
$\mathbf{x}^{\mathrm{a}}$ = vector of nonzero-lower-bounded local variables
$\mathbf{x}^{\mathrm{f}}$ = vector of arbitrarily-forced values
$\mathbf{x}^{\mathrm{g}}$ = vector of global variables
x_j = disaggregated variable
$\mathbf{x}^{l}$ = vector of solutions
$\mathbf{x}^{\mathrm{s}}$ = vector of zero-lower-bounded local variables
xLO = nonzero scalar, representing lower bound
xUP = nonzero scalar, representing upper bound
X = conversion

y = binary variable
Y = Boolean variable
z = continuous variable
Z = objective variable
α = one-dimensional variable
γ_j = cost coefficient (k$)
$\Delta_r H$ = reaction enthalpy (kJ)
ΔT_{ijkl} = match temperature difference for heat exchangers (K)
δ = scalar used to vary a variable between its bounds
λ = continuous variable
Φ = Heat flow rate (kW)
Φ_x = implicit function in relation equation
Ω = propositional logical constraints
$\nabla_x h(\mathbf{x}^l)$ = gradient of $h(\mathbf{x})$ in lth point

Abbreviations

ACH = alternative convex hull
CCH = conventional convex hull
CSTR = continuous stirred tank reactor
DAE = differential algebraic equation
GDP = generalized disjunctive programming
HEN = heat exchanger network
LP = linear programming problem
MILP = mixed-integer linear programming
MINLP = mixed-integer nonlinear programming
NLP = nonlinear programming
OA = outer approximation
OA/ER = outer approximation/equality relaxation
M/D = modeling/decomposition
PFR = plug-flow reactor

Superscripts

a = alternative
f = arbitrarily-forced
g = Global
in = inlet
LO = lower bound
out = outlet
n,m = dimensions of vectors
r = relation
s = substitute
UP = upper bound

Subscripts

i = term in disjunction
k = disjunction
v = variable
f = fixed
o = operational

Literature Cited

1. Grossmann IE, Kravanja Z. *Mixed-integer nonlinear programming: a survey of algorithms and applications*. In: Biegler LT, editor. *Large-Scale Optimization with Applications. Part 2: Optimal Design and Control*, Vol. 93. New York: Springer-Verlag, 1997: 73–100.
2. Biegler LT, Grossmann IE. Retrospective on optimization. *Comput Chem Eng*. 2004;28:1169–1192.
3. Grossmann IE, Biegler LT. Part II. Future perspective on optimization. *Comput Chem Eng*. 2004;28:1193–1218.
4. Raman R, Grossmann IE. Modeling and computational techniques for logic-based integer programming. *Comput Chem Eng*. 1994; 18:563–578.
5. Balas E. *Disjunctive Programming: Properties of the Convex Hull of Feasible Points*. Pittsburgh: Carnegie Mellon University, 1974. Technical report MSRR 348.
6. Jackson JR, Grossmann IE. A disjunctive programming approach for the optimal design of reactive distillation columns. *Comput Chem Eng*. 2001;25:1661–1673.
7. Jackson JR, Grossmann IE. High-level optimization model for the retrofit planning of process networks. *Ind Eng Chem Res*. 2002; 41:3762–3770.
8. Montagna JM, Iribarren OA, Vecchietti AR. Synthesis of biotechnological processes using generalized disjunctive programming. *Ind Eng Chem Res*. 2004;43:4220–4232.
9. Caballero JA, Milan-Yanez D, Grossmann IE. Rigorous design of distillation columns: integration of disjunctive programming and process simulators. *Ind Eng Chem Res*. 2005;44:6760–6775.
10. Caballero JA, Odjo A, Grossmann IE. Flowsheet optimization with complex cost and size functions using process simulators. *AIChE J*. 2007;53:2351–2366.
11. Lee S, Grossmann IE. New algorithms for nonlinear generalized disjunctive programming. *Comput Chem Eng*. 2000;24:2125–2141.
12. Turkay M, Grossmann IE. Tight mixed-integer optimization models for the solution of linear and nonlinear. *Comput Chem Eng*. 1998; 22:1229–1239.
13. Grossmann IE, Lee S. Generalized convex disjunctive programming: nonlinear convex hull relaxation. *Comput Optim Appl*. 2003;26:83–100.
14. Stein O, Oldenburg J, Marquardt W. Continuous reformulations of discrete-continuous optimization problems. *Comput Chem Eng*. 2004;28:1951–1966.
15. Lee S, Grossmann IE. Logic-based modeling and solution of nonlinear discrete/continuous optimization problems. *Ann Oper Res*. 2005;139:267–288.
16. Lee S, Grossmann IE. Global optimization of nonlinear generalized disjunctive programming with bilinear equality constraints: applications to process networks. *Comput Chem Eng*. 2003;27:1557–1575.
17. Bergamini ML, Aguirre P, Grossmann I. Logic-based outer approximation for globally optimal synthesis of process networks. *Comput Chem Eng*. 2005;29:1914–1933.
18. Sawaya NW, Grossmann IE. Computational implementation of nonlinear convex hull reformulation. *Comput Chem Eng*. 2007;31:856–866.
19. Sawaya N. Reformulations, Relaxations and Cutting Planes for Generalized Disjunctive Programming, Ph.D. Thesis. Pittsburgh, PA: Carnegie Mellon University, 2006.
20. Turkay M, Grossmann IE. Disjunctive programming techniques for the optimization of process systems with discontinuous investment costs multiple size regions. *Ind Eng Chem Res*. 1996;35:2611–2623.
21. Vecchietti A, Lee S, Grossmann IE. Modeling of discrete/continuous optimization problems: characterization and formulation of disjunctions and their relaxations. *Comput Chem Eng*. 2003;27:433–448.
22. Sawaya NW, Grossmann IE. A cutting plane method for solving linear generalized disjunctive programming problems. *Comput Chem Eng*. 2005;29:1891–1913.
23. GAMSWorld. MINLP World—Solvers. Available at: http://www.gamsworld.org/minlp/solvers.htm. Accessed: December 2007.
24. Vecchietti A, Grossmann IE. LOGMIP: a disjunctive 0–1 nonlinear optimizer for process systems models. *Comput Chem Eng*. 1997;21: 427–432.
25. Kravanja Z, Grossmann IE. New developments and capabilities in prosyn: an automated topology and parameter process synthesizer. *Comput Chem Eng*. 1994;18:1097–1114.
26. Turkay M, Grossmann IE. Logic-based MINLP algorithms for the optimal synthesis of process networks. *Comput Chem Eng*. 1996;20: 959–978.
27. Kocis GR, Grossmann IE. A modeling and decomposition strategy for the minlp optimization of process flowsheets. *Comput Chem Eng*. 1989;13:797–819.
28. Sorsak A, Kravanja Z. Simultaneous MINLP synthesis of heat exchanger networks comprising different exchanger types. *Comput Chem Eng*. 2002;26:599–615.
29. 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.
30. Yee TF, Grossmann IE. Simultaneous-optimization models for heat integration. Heat-exchanger network synthesis. *Comput Chem Eng*. 1990;14:1165–1184.

Appendix A: Calculus for Example 1

CCH:

$\lambda_1 = 1, \lambda_2 = 0$:	$0.7 \le x_1^s \le 1.0$	$x_2^s = 0$	$0.7 \le x \le 1.0$
	$10 \le z_1^s \le 20$	$z_2^s = 0$	$10 \le z \le 20$
$\lambda_1 = 0.67, \lambda_2 = 0.33$:	$0.47 \le x_1^s \le 0.67$	$0.07 \le x_2^s \le 0.17$	$0.54 \le x \le 0.84$
	$6.7 \le z_1^s \le 13.3$	$13.3 \le z_2^s \le 16.7$	$20 \le z \le 30$
$\lambda_1 = 0.33, \lambda_2 = 0.67$:	$0.23 \le x_1^s \le 0.33$	$0.13 \le x_2^s \le 0.34$	$0.36 \le x \le 0.67$
	$3.3 \le z_1^s \le 6.7$	$26.7 \le z_2^s \le 33.3$	$30 \le z \le 40$
$\lambda_1 = 1, \lambda_2 = 0$:	$x_1^s = 0$	$0.2 \le x_2^s \le 0.5$	$0.2 \le x \le 0.5$
	$z_1^s = 0$	$40 \le z_2^s \le 50$	$40 \le z \le 50$

ACH:

$\lambda_1 = 1, \lambda_2 = 0$:	$0.7 \le x_1^a \le 1.0$	$x_2^a = 0.2$	$0.7 \le x \le 1.0$
	$10 \le z_1^a \le 20$	$z_2^a = 40$	$10 \le z \le 20$
$\lambda_1 = 0.67, \lambda_2 = 0.33$:	$0.7 \le x_1^a \le 0.9$	$0.2 \le x_2^a \le 0.3$	$0.54 \le x \le 0.84$
	$10 \le z_1^a \le 16.7$	$40 \le z_2^a \le 43.3$	$20 \le z \le 30$
$\lambda_1 = 0.33, \lambda_2 = 0.67$:	$0.7 \le x_1^a \le 0.8$	$0.2 \le x_2^a \le 0.4$	$0.36 \le x \le 0.67$
	$10 \le z_1^a \le 13.3$	$40 \le z_2^a \le 46.7$	$30 \le z \le 40$
$\lambda_1 = 1, \lambda_2 = 0$:	$x_1^a = 0.7$	$0.2 \le x_2^a \le 0.5$	$0.2 \le x \le 0.5$
	$z_1^a = 10$	$40 \le z_2^a \le 50$	$40 \le z \le 50$

Appendix B: Proofs of Obtaining the Alternative Formulations of Outer Approximations

Directly from (CCH-MILP):

Substituting $\mathbf{x}^s$ by $\mathbf{x}^a - \mathbf{x}^f(1 - Y_{ik})$ in the linearization of (CCH-MILP)

$$\nabla_x h_{ik}(\mathbf{x}^l)^T \mathbf{x}^s \le \left(\nabla_x h_{ik}(\mathbf{x}^l)^T \mathbf{x}^l - h_{ik}(\mathbf{x}^l)\right) y_{ik} \tag{B1}$$

gives:

$$\nabla_x h_{ik}(\mathbf{x}^l)^T \left(\mathbf{x}^a - \mathbf{x}^f(1 - y_{ik})\right) \le \left(\nabla_x h_{ik}(\mathbf{x}^l)^T \mathbf{x}^l - h_{ik}(\mathbf{x}^l)\right) y_{ik} \tag{B2}$$

After rearrangement we obtain:

$$\nabla_x h_{ik}(\mathbf{x}^l)^T x^a \le \nabla_x h_{ik}(\mathbf{x}^l)^T x^f + \left(\nabla_x h_{ik}(\mathbf{x}^l)^T (\mathbf{x}^l - \mathbf{x}^f) - h_{ik}(\mathbf{x}^l)\right) y_{ik} \tag{B3}$$

Directly from (A-CHRP):

By considering the linearization

$$\nabla_x h_{ik}(\mathbf{x}^l)^T x^s - \left(\nabla_x h_{ik}(\mathbf{x}^l)^T \mathbf{x}^l - h_{ik}(\mathbf{x}^l)\right) \le 0 \tag{B4}$$

as g_{ik} in the nonlinear constraint of problem (A-CHRP):

$$\lambda_{ik} g_{ik}\left(\frac{x^a - x^f(1 - \lambda_{ik})}{\lambda_{ik}}\right) \le 0 \tag{B5}$$

where a continuous variable λ_{ik} is replaced by a binary variable Y_{ik}, the following inequality is obtained:

$$y_{ik} \frac{\nabla_x h_{ik}(\mathbf{x}^l)^T \left(\mathbf{x}^a - \mathbf{x}^f(1 - y_{ik})\right)}{y_{ik}} + y_{ik} \cdot \left(-\nabla_x h_{ik}(\mathbf{x}^l)^T \mathbf{x}^l + h_{ik}(\mathbf{x}^l)\right) \le 0 \tag{B6}$$

or

$$\nabla_x h_{ik}(\mathbf{x}^l)^T \left(\mathbf{x}^a - \mathbf{x}^f(1 - y_{ik})\right) + \left(-\nabla_x h_{ik}(\mathbf{x}^l)^T \mathbf{x}^l + h_{ik}(\mathbf{x}^l)\right) y_{ik} \le 0 \tag{B7}$$

and finally:

$$\nabla_x h_{ik}(\mathbf{x}^l)^T \mathbf{x}^a \le \nabla_x h_{ik}(\mathbf{x}^l)^T \mathbf{x}^f + \left(\nabla_x h_{ik}(\mathbf{x}^l)^T (\mathbf{x}^l - \mathbf{x}^f) - h_{ik}(\mathbf{x}^l)\right) y_{ik} \tag{B8}$$

which is the same as (B3).

By a convex hull of linearizations from selected and rejected alternatives in (A-OAMP):

When an alternative is selected ($Y_{ik} = 1$), the following linearization is applied:

$$\nabla_x h_{ik}(\mathbf{x}^l)^T \mathbf{x}^a \le \nabla_x h_{ik}(\mathbf{x}^l)^T \mathbf{x}^l - h_{ik}(\mathbf{x}^l) \tag{B9}$$

while when the alternative is rejected ($1 - Y_{ik}$), its auxiliary constraint, which is not shown in (A-OAMP) since it is redundant:

$$\nabla_x h_{ik}(\mathbf{x}^l)^T \mathbf{x}^a \le \nabla_x h_{ik}(\mathbf{x}^l)^T \mathbf{x}^f \tag{B10}$$

has to be satisfied to preserve feasibility. A convex hull of Eqs. B9 and B10 is then given by:

$$\nabla_x h_{ik}(\mathbf{x}^l)^T \mathbf{x}^a \le \left(\nabla_x h_{ik}(\mathbf{x}^l)^T \mathbf{x}^l - h_{ik}(\mathbf{x}^l)\right) y_{ik} + \nabla_x h_{ik}(\mathbf{x}^l)^T \mathbf{x}^f (1 - y_{ik}) \tag{B11}$$

or:

$$\nabla_x h_{ik}(\mathbf{x}^l)^T \mathbf{x}^a \le \nabla_x h_{ik}(\mathbf{x}^l)^T \mathbf{x}^f + \left(\nabla_x h_{ik}(\mathbf{x}^l)^T (\mathbf{x}^l - \mathbf{x}^f) - h_{ik}(\mathbf{x}^l)\right) y_{ik} \tag{B12}$$

which is the same as B3.

Appendix C: Detailed Solution Procedure for Example 2[27]

Step 1—NLP1: The first initial point is selected, i.e., $(y_1, y_2) = (0,1)$. The NLP subproblem to be solved at iteration 1 is given as:

$$\begin{aligned} \min\ & z = 5.5 + 6V_2 + 5x \\ \text{s.t.}\ & z_2 = 0.8[1 - \exp(-0.4)]x_2 \\ & z_1 + z_2 = 10 \\ & x_1 + x_2 = x \\ & V_1 = 0,\ x_1 = 0,\ z_1 = 0 \\ & V_2^{LO}, x_2^{LO}, z_2^{LO} \le V_2, x_2, z_2 \le V_2^{UP}, x_2^{UP}, z_2^{UP} \end{aligned} \quad \text{(SNLP1)}$$

The solution of this NLP is $z = 107.376$ at $x = x_2 = 15$ and $V_2 = 4.479$ and the Lagrange multiplier for the mixer mass balance ($z_1 + z_2 = 10$) is $\mu = -7.5$.

Step 2—Suboptimization: The next step is to perform the suboptimization of the nonexisting reactor using the Lagrangean suboptimization scheme. The feed stream is fixed at the optimal value of the splitter inlet stream in the above NLP problem ($x = 15$) and the Lagrange multiplier μ for the mixer mass balance is used to derive a price for the reactor outlet stream (z_1). The resulting suboptimization problem is then given as:

$$\begin{aligned} \min\ & z = 7.5 + 7V_1 + 5x_1 - 7.5z_1 \\ \text{s.t.}\ & z_1 = 0.9[1 - \exp(-0.5)]x_1 \\ & x_1 = 15 \\ & V_1 \le 0 \\ & V_1^{LO}, x_1^{LO}, z_1^{LO} \le V_1, x_1, z_1 \le V_1^{UP}, x_1^{UP}, z_1^{UP} \end{aligned} \quad \text{(SNLP2)}$$

The solution of the above suboptimization problem yields $z_1 = 11.633$ and $V_1 = 3.957$.

Step 3—MILP1: By deriving big-M, convex hull and alternative convex hull linearizations at the above solution point and appropriately considering the sign of the Lagrangean multipliers the following outer-approximations for the MILP master problem have been obtained:

$$\begin{aligned} \text{Big-M:}\ & z_1 \le 0.776x_1 + 0.933V_1 - 3.693 + M(1 - y_1) \\ & z_2 \le 0.666x_2 + 0.800V_2 - 3.584 + M(1 - y_2) \\ \text{CCH:}\ & z_1^s \le 0.776x_1^s + 0.933V_1^s - 3.693y_1 \\ & z_2^s \le 0.666x_2^s + 0.800V_2^s - 3.584y_2 \\ \text{ACH:}\ & z_1^a \le 0.776x_1^a + 0.933V_1^a - 0.744 - 2.949y_1 \\ & z_2^a \le 0.666x_2^a + 0.800V_2^a - 0.067 - 3.650y_2 \end{aligned}$$

The complete MILP master problems for the Big-M, CCH, and ACH representations are given below, respectively:

$$\begin{aligned} \min\ & z = 7.5y_1 + 5.5y_2 + 7V_1 + 6V_2 + 5x \\ \text{s.t.}\ & z_1 \le 0.776x_1 + 0.933V_1 - 3.693 + M(1 - y_1) \\ & z_2 \le 0.666x_2 + 0.800V_2 - 3.584 + M(1 - y_2) \\ & z_1 + z_2 = 10 \\ & x_1 + x_2 = x \\ & V_1 \le V_1^{UP}y_1 \quad V_2 \le V_2^{UP}y_2 \\ & V_1 \ge VLO_1y_1 \quad V_2 \ge VLO_2y_2 \\ & x_1 \le x_1^{UP}y_1 \quad x_2 \le x_2^{UP}y_2 \\ & x_1 \ge xLO_1y_1 \quad x_2 \ge xLO_2y_2 \\ & z_1 \le z_1^{UP}y_1 \quad z_2 \le z_2^{UP}y_2 \\ & z_1 \ge zLO_1y_1 \quad z_2 \ge zLO_2y_2 \\ & y_1 + y_2 = 1 \\ & 0 \le V_1, x_1, z_1, V_2, x_2, z_2 \le V_1^{UP}, x_1^{UP}, z_1^{UP}, V_2^{UP}, x_2^{UP}, z_2^{UP} \end{aligned} \quad \text{(Big-M MILP)}$$

(CCH MILP):

$$\begin{aligned} \min\ & z = 7.5y_1 + 5.5y_2 + 7V_1^s + 6V_2^s + 5x \\ \text{s.t.}\ & z_1^s \le 0.776x_1^s + 0.933V_1^s - 3.693y_1 \\ & z_2^s \le 0.666x_2^s + 0.800V_2^s - 3.584y_2 \\ & z_1^s + z_2^s = 10 \\ & x_1^s + x_2^s = x \\ & V_1^s \le VUP_1y_1 \\ & V_2^s \le VUP_2y_2 \\ & V_1^s \ge VLO_1y_1 \\ & V_2^s \ge VLO_2y_2 \\ & x_1^s \le xUP_1y_1 \\ & x_2^s \le xUP_2y_2 \\ & x_1^s \ge xLO_1y_1 \\ & x_2^s \ge xLO_2y_2 \\ & z_1^s \le zUP_1y_1 \\ & z_2^s \le zUP_2y_2 \\ & z_1^s \ge zLO_1y_1 \\ & z_2^s \ge zLO_2y_2 \\ & y_1 + y_2 = 1 \end{aligned}$$

$$\xrightarrow[x^f = x^{lo}]{x^s = x^a - x^f(1 - y)}$$

(ACH MILP):

$$\begin{aligned} \min\ & z = 7.5y_1 + 5.5y_2 + 7V_1^a + 6V_2^a + 5x \\ & \quad - 7V_1^{lo}(1 - y_1) - 6V_2^{lo}(1 - y_2) \\ \text{s.t.}\ & z_1^a \le 0.776x_1^a + 0.933V_1^a - 0.744 - 2.949y_1 \\ & z_2^a \le 0.666x_2^a + 0.800V_2^a + 0.067 - 3.650y_2 \\ & z_1^a + z_2^a - z_1^{lo}(1 - y_1) - z_2^{lo}(1 - y_2) = 10 \\ & x_1^a + x_2^a - x_1^{lo}(1 - y_1) - x_2^{lo}(1 - y_2) = x \\ & V_1^a \le V_1^{lo} + (VUP_1 - V_1^{lo})y_1 \\ & V_2^a \le V_2^{lo} + (VUP_2 - V_2^{lo})y_2 \\ & x_1^a \le x_1^{lo} + (xUP_1 - x_1^{lo})y_1 \\ & x_2^a \le x_2^{lo} + (xUP_2 - x_2^{lo})y_2 \\ & z_1^a \le z_1^{lo} + (zUP_1 - z_1^{lo})y_1 \\ & z_2^a \le z_2^{lo} + (zUP_2 - z_2^{lo})y_2 \\ & y_1 + y_2 = 1 \\ & V_1^{lo}, x_1^{lo}, z_1^{lo}, V_2^{lo}, x_2^{lo}, z_2^{lo} \le V_1, x_1, z_1, V_2, x_2, z_2 \end{aligned}$$

where M is a big scalar and VLO, XLO, and zLO are nonzero scalars representing the variable lower bounds. The solution of the resulting master problem is the same for all the three formulations: $z = 95.78$ at the point $(y_1,y_2) = (1,0)$. The ACH formulation has the smallest number of rows.

Step 4—Iteration 2: The solution to the second NLP subproblem at $(y_1,y_2) = (1,0)$ yields $z = 99.24$. The MILP master problem at iteration 2 is infeasible since both reactors have already been examined, and the OA/ER algorithm terminates.

Manuscript received June 19, 2008, and revision received Mar. 10, 2009.