Translation of variables for different applications in process synthesis

Marcel Ropotar[a,b,+], *Zdravko Kravanja*[a,*]

[a]*University of Maribor, Faculty of Chemistry and Chemical Engineering, P.O. Box 219, 2000 Maribor/Slovenija,*

[b]*Tanin Sevnica kemična industrija d.d., Hermanova cesta 1, Sevnica, SI - 8290 Slovenija*

[+]marcel.ropotar@uni-mb.si

[*]zdravko.kravanja@uni-mb.si

Abstract

The aim of this contribution is to present a special translation of variables, which can be applied to logic-based and mixed-integer programming problems in process synthesis. Our research focused on the development of an alternative convex-hull representation for a logic-based outer-approximation algorithm (OA), implemented in an automated process synthesizer MIPSYN. Three examples were solved in order to compare models with and without translation of variables.

1. Introduction

Recent developments in logic-based optimization (e.g. Grossmann and Biegler, 2004) are regarded as some of the most important achievements for effectively modeling and solving discrete-continuous synthesis problems. One of the possible representations of discrete-continuous problems is the Generalized Disjunctive Programming (GDP), which was developed by Raman and Grossmann (1994) as an extension of the disjunctive programming paradigm developed by Balas (1974). GDP problems could be solved either by transforming it into mixed-integer programs or by the development of specific solution methods, e.g. the branch and bound algorithm with convex relaxation by Lee and Grossmann, 2000. Sawaya (2006) showed that the tightness of convex hull representation of disjunctions could be significantly improved by moving global constraints into representation of disjunctions.

In this contribution we present another idea – a variable translation in order to have a narrower space of variables, which may then further increases efficiency when solving discrete-continuous problems.

2. Variable translation

In synthesis problems, continuous variables v^{s} are usually defined within zero lower and non-zero upper bounds and constrained by

$$v^{\mathrm{LO}}y \le v^{\mathrm{s}} \le v^{\mathrm{LO}}y \tag{1}$$

in order to force non-zero bounds when alternatives are selected ($y = 1$). The main idea is to substitute a zero-lower-bounded variable v^{s} of alternatives ($0 \le v^{\mathrm{s}} \le v^{\mathrm{s,UP}}$) by a non-zero-lower-bounded variable ($v^{\mathrm{LO}} \le v \le v^{\mathrm{UP}}$), through the use of the following translation equation:

$$v^{\mathrm{s}} = v - v^{\mathrm{f}}(1 - y) \tag{2}$$

where v^{f} is an arbitrarily-forced scalar from the interval (v^{LO},v^{UP}) and y is a corresponding binary variable. When an alternative is selected, an integer term $v^{\mathrm{f}}(1 - y)$ becomes zero and v^{s} becomes equal to v, and when it is rejected, a value v^{f} is subtracted from the variable v. When eq. (2) is applied to eq. (1) we obtain

$$v^{\mathrm{f}} + (v^{\mathrm{LO}} - v^{\mathrm{f}})y \leq v \leq v^{\mathrm{f}} + (v^{\mathrm{UP}} - v^{\mathrm{f}})y \tag{3}$$

Note that when $y = 1$, it follows that v is constrained within its non-zero bounds, and when $y = 0$, v becomes v^{f}. In this way the original space of variable is preserved irrespective of discrete decisions. Intuitively, one could expect that retaining within the narrower original space of variables would increase the efficiency of the GDP. Two additional types were obtained besides a mixed integer type of variable translation (eq. (2)). The following relaxed translation formula is obtained if a binary variable y is relaxed to a continuous variable λ defined between 0 and 1.

$$v^{\mathrm{s}} = v - v^{\mathrm{f}}(1 - \lambda) \tag{4}$$

A logic-based form of the variable translation is:

$$[Y\colon v^{\mathrm{s}} = v] \vee [\neg Y\colon v^{\mathrm{s}} = v - v^{\mathrm{f}}] \tag{5}$$

for a Boolean variable $Y = true$ it follows that $v^{\mathrm{s}} = v$ and for $Y = false$ $v^{\mathrm{s}} = v - v^{\mathrm{f}}$.

2.1. Alternative logic-based OA algorithm and MILP transformation

Selection of a different variable translation type depends on a type of model (MIP – eq. (2), relaxed MIP – eq. (4), logic-based – eq. (5)).

A convex hull representation is the tightest relaxation of disjunctions in Generalized Disjunctive Programming − GDP problem. It is generated from taking the linear combination of all points in feasible regions of disjunctions. By applying the convex hull relaxation to the GDP problem the (CHRP) problem given bellow is obtained. Since the problem is relaxed, the relaxed form of the variable translation (eq. (4)) is applied in order to translate its zero-lower-bounded variables into nonzero-lower-bounded variables. The following alternative (A-CHRP) problem is obtained:

(CHRP):

$$\min Z = \sum_k \sum_i \left(\gamma_{ik}\lambda_{ik} + f^{\mathrm{a}}_{ik}\left(v_{ik}\right)\right) + f^{\mathrm{g}}\left(\boldsymbol{x}\right)$$

$$\begin{aligned}
\text{s.t.} \quad & h(\boldsymbol{x}) \le 0 \\
& A^{\mathrm{g}}(\boldsymbol{x}) \le b^{\mathrm{g}} \\
& \boldsymbol{x} = \sum_{i \in D_k} v_{ik}, \quad k \in SD \\
& \sum_{i \in D_k} \lambda_{ik} = 1, \quad 0 \le \lambda_{ik} \le 1 \\
& E\boldsymbol{\lambda} \le e \\
& \lambda_{ik} VLO_{ik} \le v_{ik}, \quad i \in D_k, k \in SD \\
& v_{ik} \le \lambda_{ik} v^{\mathrm{UP}}_{ik}, \quad i \in D_k, k \in SD \\
& \lambda_{ik} g_{ik}\left(\frac{v_{ik}}{\lambda_{ik}}\right) \le 0, \quad i \in D_k, k \in SD \\
& \boldsymbol{x}^{\mathrm{LO}} \le \boldsymbol{x} \le \boldsymbol{x}^{\mathrm{UP}} \\
& 0 \le v_{ik} \le v^{\mathrm{UP}}_{ik}, \quad i \in D_k, k \in SD
\end{aligned}$$

$$\xrightarrow{v^{\mathrm{s}} = v - v^{\mathrm{f}}(1-\lambda)}$$

(A-CHRP):

$$\min Z = \sum_k \sum_i \left(\gamma_{ik}\lambda_{ik} + f^{\mathrm{a}}_{ik}\left(v_{ik} - v^{\mathrm{f}}_{ik}\left(1-\lambda_{ik}\right)\right)\right) + f^{\mathrm{g}}\left(\boldsymbol{x}\right)$$

$$\begin{aligned}
\text{s.t.} \quad & h(\boldsymbol{x}) \le 0 \\
& A^{\mathrm{g}}(\boldsymbol{x}) \le b^{\mathrm{g}} \\
& \boldsymbol{x} = \sum_{i \in D_k} v_{ik} - v^{\mathrm{f}}_{ik}\left(1-\lambda_{ik}\right), k \in SD \\
& \sum_{i \in D_k} \lambda_{ik} = 1, \quad 0 \le \lambda_{ik} \le 1 \\
& E\boldsymbol{\lambda} \le e \\
& v^{\mathrm{LO}}_{ik}\lambda_{ik} + v^{\mathrm{f}}_{ik}\left(1-\lambda_{ik}\right) \le v_{ik}, \quad i \in D_k, k \in SD \\
& v_{ik} \le v^{\mathrm{UP}}_{ik}\lambda_{ik} + v^{\mathrm{f}}_{ik}\left(1-\lambda_{ik}\right), \quad i \in D_k, k \in SD \\
& \lambda_{ik} g_{ik}\left(\frac{v_{ik} - v^{\mathrm{f}}_{ik}\left(1-\lambda_{ik}\right)}{\lambda_{ik}}\right) \le 0, i \in D_k, k \in SD \\
& \boldsymbol{x}^{\mathrm{LO}} \le \boldsymbol{x} \le \boldsymbol{x}^{\mathrm{UP}} \\
& v^{\mathrm{LO}}_{ik} \le v_{ik} \le v^{\mathrm{UP}}_{ik}, \quad i \in D_k, k \in SD
\end{aligned}$$

where v_{ik} are disaggregated variables and VLO_{ik} nonzero scalars in the problem (CHRP) forcing the nonzero lower bounds when an alternative is selected. Note that in the alternative formulation (A-CHRP), nonzero lower bounds can now be applied directly to the disaggregated variables v_{ik}. In both convex hull representations, global variables can in principle be defined between nonzero lower and upper bounds. However, in the conventional (CHRP) problem, disaggregated variables should always have zero lower bounds so that they can obtain zero values in the bounding constraints when λ_{ik} becomes zero. In the case of the alternative problem (A-CHRP), when λ_{ik} takes zero value, the corresponding variables v_{ik} in the bounding constraints are set to v^{f}_{ik}. At the same time the terms v_{ik} *and* $v^{\mathrm{f}}_{ik}\left(1-\lambda_{ik}\right)$ in the balance equation and the objective function precisely cancel each other out, which is equivalent to obtaining zero values for v_{ik} in the original problem (CHRP). Note that if the continuous variables λ_{ik} are replaced by integer variables y_{ik}, MINLP reformulation is obtained.

Lee and Grossmann (2000) showed that applying the outer-approximation method to the MINLP reformulation of the convex hull relaxation regarding problem (GDP) reduces to the logic-based OA method by Turkay and Grossmann (1996). Logic-based OA problems are usually solved through MILP transformation where Boolean variables Ys are replaced by binary variables ys, logical relations are formulated as integer constrains and disjunctives are represented either by big-M or 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 translation of variables, using eq. (2), the following alternative MILP master problem (ACH-MILP) is obtained:

(CCH-MILP):

$$\min\ Z = \sum_i \sum_k \left(c_{ik}\, y_{ik} + \alpha_{ik}^{\mathrm{a}}\right) + \alpha^{\mathrm{g}}$$

s.t.

$$\left.\begin{array}{l} \alpha^{\mathrm{g}} \ge f\left(\boldsymbol{x}^l\right) + \nabla_x f\left(\boldsymbol{x}^l\right)^{\mathrm{T}} (\boldsymbol{x} - \boldsymbol{x}^l) \\ h\left(\boldsymbol{x}^l\right) + \nabla_x h\left(\boldsymbol{x}^l\right)^{\mathrm{T}} (\boldsymbol{x} - \boldsymbol{x}^l) \le 0 \end{array}\right\},\quad l = 1,\ldots,L$$

$$A^{\mathrm{g}}(\boldsymbol{x}) \le b^{\mathrm{g}}$$

$$E^{\mathrm{g}}(\boldsymbol{y}) \le e^{\mathrm{g}}$$

$$\boldsymbol{xLO} y_{ik} \le \boldsymbol{x}$$

$$\boldsymbol{x} \le \boldsymbol{x}^{\mathrm{UP}} y_{ik}$$

$$A^{ik}\left(\boldsymbol{x}\right) \le b_{ik} y_{ik}$$

$$\nabla_x f_{ik}^{\mathrm{a}}\left(\boldsymbol{x}^l\right)^{\mathrm{T}} \boldsymbol{x} - \alpha_{ik}^{\mathrm{a}} \le \left[\nabla_x f_{ik}^{\mathrm{a}}\left(\boldsymbol{x}^l\right)^{\mathrm{T}} \boldsymbol{x}^l - f_{ik}^{\mathrm{a}}\left(\boldsymbol{x}^l\right)\right] y_{ik}$$

$$\nabla_x h_{ik}\left(\boldsymbol{x}^l\right)^{\mathrm{T}} \boldsymbol{x} \le \left[\nabla_x h_{ik}\left(\boldsymbol{x}^l\right)^{\mathrm{T}} \boldsymbol{x}^l - h_{ik}\left(\boldsymbol{x}^l\right)\right] y_{ik}$$

$$x \in R^n,\ y_{ik} \in \{0,1\}^m$$

$$0 \le \alpha^{\mathrm{g}}, \alpha_{ik}^{\mathrm{a}} \qquad i \in D_k, k \in SD$$

$$0 \le \boldsymbol{x} \le \boldsymbol{x}^{\mathrm{UP}}$$

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

(ACH-MILP):

$$\min\ Z = \sum_i \sum_k \left(c_{ik} y_{ik} + \alpha_{ik}^{\mathrm{a}}\right) + \alpha^{\mathrm{g}}$$

s.t.

$$\left.\begin{array}{l} \alpha^{\mathrm{g}} \ge f\left(\boldsymbol{x}^l\right) + \nabla_x f\left(\boldsymbol{x}^l\right)^{\mathrm{T}} (\boldsymbol{x} - \boldsymbol{x}^l) \\ h\left(\boldsymbol{x}^l\right) + \nabla_x h\left(\boldsymbol{x}^l\right)^{\mathrm{T}} (\boldsymbol{x} - \boldsymbol{x}^l) \le 0 \end{array}\right\},\quad l = 1,\ldots,L$$

$$A^{\mathrm{g}}(\boldsymbol{x}) \le b^{\mathrm{g}}$$

$$E^{\mathrm{g}}(\boldsymbol{y}) \le e^{\mathrm{g}}$$

$$\boldsymbol{x}^{\mathrm{f}} + (\boldsymbol{x}^{\mathrm{LO}} - \boldsymbol{x}^{\mathrm{f}}) y_{ik} \le \boldsymbol{x} \qquad (6)$$

$$\boldsymbol{x} \le \boldsymbol{x}^{\mathrm{f}} + (\boldsymbol{x}^{\mathrm{UP}} - \boldsymbol{x}^{\mathrm{f}}) y_{ik} \qquad (7)$$

$$A^{ik}\left(\boldsymbol{x} - \boldsymbol{x}^{\mathrm{f}}\left(1 - y_{ik}\right)\right) \le b_{ik} y_{ik}$$

$$\nabla_x f_{ik}^{\mathrm{a}}\left(\boldsymbol{x}^l\right)^{\mathrm{T}} \boldsymbol{x} - \alpha_{ik}^{\mathrm{a}} \le \nabla_x f_{ik}^{\mathrm{a}}\left(\boldsymbol{x}^l\right)^{\mathrm{T}} \boldsymbol{x}^{\mathrm{f}} +$$
$$\left[\nabla_x f_{ik}^{\mathrm{a}}\left(\boldsymbol{x}^l\right)^{\mathrm{T}} (\boldsymbol{x}^l - \boldsymbol{x}^{\mathrm{f}}) - f_{ik}^{\mathrm{a}}\left(\boldsymbol{x}^l\right)\right] y_{ik}$$

$$\nabla_x h_{ik}\left(\boldsymbol{x}^l\right)^{\mathrm{T}} \boldsymbol{x} \le \nabla_x h_{ik}\left(\boldsymbol{x}^l\right)^{\mathrm{T}} \boldsymbol{x}^{\mathrm{f}} + \qquad (8)$$
$$\left[\nabla_x h_{ik}\left(\boldsymbol{x}^l\right)^{\mathrm{T}} (\boldsymbol{x}^l - \boldsymbol{x}^{\mathrm{f}}) - h_{ik}\left(\boldsymbol{x}^l\right)\right] y_{ik}$$

$$x \in R^n,\ y_{ik} \in \{0,1\}^m$$

$$0 \le \alpha^{\mathrm{g}}, \alpha_{ik}^{\mathrm{a}} \qquad i \in D_k, k \in SD$$

$$\boldsymbol{x}^{\mathrm{LO}} \le \boldsymbol{x} \le \boldsymbol{x}^{\mathrm{UP}}$$

where L, SD and D_k are sets of NLP solutions, disjunctives and terms in disjunctives, respectively. The key feature of the alternative outer approximations (ineq. (8)) is that they preserve feasibility when alternatives are not selected when $\boldsymbol{x}$ is set to $\boldsymbol{x}^{\mathrm{f}}$. This enables the use of variables with non-zero lower bounds. Note that if $\boldsymbol{x}^{\mathrm{f}}$ is set to $\boldsymbol{x}^{\mathrm{LO}}$, ineq. (6) becomes redundant, if it is set to $\boldsymbol{x}^{\mathrm{UP}}$, ineq. (7) becomes redundant. If the lower bounds are zero, the problem (ACH-MILP) reduces to the problem (CCH-MILP). The MILP model given above is implemented in the MINLP process synthesizer MIPSYN.

2.2. Implementation in a process synthesizer MIPSYN

Until recently only big-M models were used in MIPSYN, the successor of PROSYN-MINLP (Kravanja and Grossmann, 1994), to solve MINLP synthesis problems. Now, the conventional convex hull and the alternative convex hull formulations, using translation of variables, are implemented in MIPSYN, too. The models are formulated in the most generalized form using various capabilities of the high-level language of GAMS. Data-and-topology independent models were developed in this way.

3. Examples

Three synthesis problems of different sizes and complexities have been solved in order to test and compare the efficiencies of models with and without translation of variables.

Example 1: The first example is a network synthesis problem with a simple model but very large-scale combinatorics with 400 binary variables. This numerical problem is an extension of the small flowsheet problem by Kocis and Grossmann (1989). Additional pairs of reactors were added to the superstructure. The objective is to minimize total cost at the fixed demand of the final outflow; $\boldsymbol{x}^{\mathrm{f}}$ was set to $\boldsymbol{x}^{\mathrm{LO}}$.

$$\min z = c_{\mathrm{f}\,i}^{1} y_i^1 + c_{\mathrm{v}}^{1} V_i^1 \;+\; c_{\mathrm{f}\,i}^{2} y_i^2 + c_{\mathrm{v}\,i}^{2} V_i^2 \;+\; c_{\mathrm{o}\,i}^{1} x_i^1 + c_{\mathrm{o}\,i}^{2} x_i^2 + 5x_1 \quad i \in 200 \tag{9}$$

The solution statistics until the third major MINLP iteration are reported in Table 1. As can be seen in Table 1, it was impossible with big-M formulation to solve the problem within a reasonable time, whilst both convex hull representations enable the solving of this high-combinatorial problem very quickly. Note that with the same integrality gap and smaller number of constraints, the alternative formulation (ACH) could solve the problem in only a quarter of the CPU time needed to solve the problem using the conventional convex hull formulation (CCH).

Table 1. MILP solution statistics of the reactor network synthesis problem.

	Best NLP	**Int. gap, %**	**No. of eq./ No.of var.**	**No. of iterations**	**No. of nodes**	**CPU for 3 it., sec.**	**Nodes/s for 3 it.**
Big-M	n/a	n/a	3802/1801	n/a	n/a	n/a	n/a
CCH	183.87	0.868	3402/1801	23214	319	19.2	16.6
ACH	183.87	0.868	2202/1801	4696	293	5.5	53.6

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

Example 2: The second example is the synthesis of a heat exchanger network (HEN) comprising different types of exchangers. Each match in a stage-wise superstructure is comprised of a double pipe, a plate and frame, a shell and tube exchanger, and a by-pass (Figure 1). Consideration of different types of exchanger enables the simultaneous selection of exchanger types; however, it significantly increases the number of binary variables. The model thus exhibits moderate complexity and high combinatorics (249 binary variables).

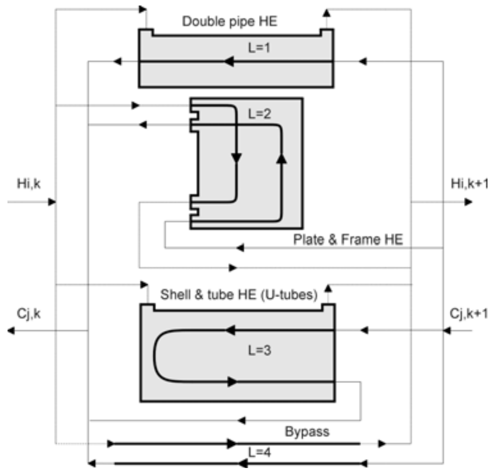

Figure 1. Match superstructure.

Table 2 shows the statistics when $\boldsymbol{x}^{\mathrm{f}}$ was set to $\boldsymbol{x}^{\mathrm{LO}}$. With respect to the integrality gap, number of iterations, CPU time and number of nodes, both convex hull representations significantly outperform the big-M one, whilst the efficiency of the alternative convex hull formulation is approximately twice that of the conventional formulation.

Table 2. MILP solution statistics for the HEN synthesis problem.

	Best NLP	**Int. gap, %**	**No. of eq./ No.of var.**	**No. of iterations**	**No. of nodes**	**CPU for 15 it., sec.**	**Nodes/s for 15 it.**
Big-M	884.07	1.548	9214/5595	2248581	70257	500.9	140.3
CCH	818.69	0.607	6894/5595	612529	35150	163.3	215.2
ACH	818.69	0.607	4574/5595	321062	19411	83.8	231.6

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

<u>Example 3:</u> The last, allyl chloride example, is the synthesis of a reactor/separator network within an overall heat integrated process scheme, with a complex model and moderate-size combinatorics (184 binary variables). The reactor/separator superstructure (Figure 2) 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. The overall model is highly nonlinear and nonconvex. Therefore, many numerical and other issues are present which makes any comparison between formulations harder, e.g. due to the effects of nonconvexities it is impossible to compare different formulations based on an integrality gap.

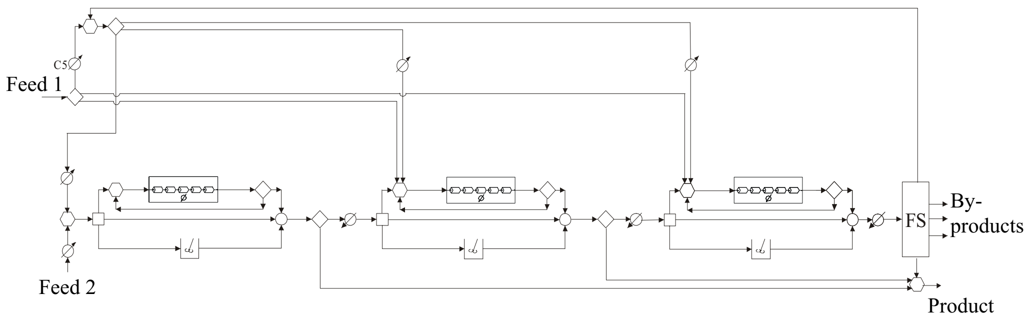

Figure 2. Reactor/separator superstructure of the allyl chloride problem.

Table 3 shows solution statistics until the 17th major MINLP iteration. As can be seen from Table 3, the efficiency of the ACH formulation is almost twice as good as that of CCH and four times better than that of Big-M. It should be noted that selection of the optimal final element in 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.

Table 3. MILP solution statistics of the allyl chloride problem

	Best NLP	Int. gap, %	No. of eq./ No.of var.	No. of iterations	No. of nodes	CPU for 17 it., sec.	Nodes/s for 17 it.
Big-M	81.924	0.190	2307/2193	4525051	19830	2166.5	9.2
CCH	81.836	100.00	1743/1011	1503644	8053	960.3	8.4
ACH	81.769	0.343	1230/1313	854049	7700	529.2	14.6

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

4. Conclusions

By the use of translation of variables on the conventional convex hull representation the alternative convex hull representation was obtained. Initial experiences indicate that the alternative convex hull representation is generally more efficient when solving high-combinatorial problems than the conventional one and has the smallest model sizes. In spite of the above mentioned efficiency, ACH formulations exhibit stronger sensitivity to the effects of nonconvexities, and the model representations are more complicated. Hence, when solving process synthesis examples the use of translation of variables on models is especially worthwhile when models are generated automatically as in the case with the equation-oriented modular synthesizer MIPSYN, since the likelihood of achieving the best efficiency of the MINLP search is increased.

References

Balas, E. (1974). *Disjunctive programming: Properties of the convex hull of feasible points*, technical report MSRR 348, Carnegie Mellon University, 1974.

Grossmann, I. E. and L. T. Biegler (2004). *Part II. Future perspective on optimization*. Comput Chem Eng, 28(8): 1193-1218.

Kocis, G. R. and I. E. Grossmann (1989). *A Modeling and Decomposition Strategy for the Minlp Optimization of Process Flowsheets*. Comput. Chem. Eng., 13(7): 797-819.

Kravanja, Z. and I. E. Grossmann (1994). *New Developments and Capabilities in Prosyn - an Automated Topology and Parameter Process Synthesizer*. Comput Chem Eng, 18(11-12): 1097-1114.

Lee, S. and I. E. Grossmann (2000). *New algorithms for nonlinear generalized disjunctive programming*. Computers and Chemical Engineering, 24: 2125-2141.

Raman, R. and I. E. Grossmann (1994). *Modeling and Computational Techniques for Logic-Based Integer programming*. Comput Chem Eng, 18(7): 563-578.

Sawaya, N. (2006). *Reformulations, relaxations and cutting planes for generalized disjunctive programming*, Thesis (Ph.D), Carnegie Mellon University, Carnegie Mellon University, 2006.

Turkay, M. and I. E. Grossmann (1996). *Logic-based MINLP algorithms for the optimal synthesis of process networks*. Comput Chem Eng, 20(8): 959-978.