16th European Symposium on Computer Aided Process Engineering
and 9th International Symposium on Process Systems Engineering
W. Marquardt, C. Pantelides (Editors)

Implementation of efficient logic-based techniques in the MINLP process synthesizer MIPSYN

Marcel Ropotar, Zdravko Kravanja

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

Abstract

The main aim of the research is to implement the most advanced modeling and solution techniques in the automated process synthesizer MIPSYN. In particular, different modeling formulations are studied, rooted in disjunctive programming and convex hull representation. Alternative modeling is proposed for logical interconnection nodes and alternative outer approximation formulation. Initial research indicates that they could be efficient for solving large-combinatorial process network problems.

Keywords: disjunctive programming, outer-approximations, MINLP, process synthesis, process synthesizer.

1. Introduction

Over the last couple of decades significant advances have been achieved in modeling and mathematical programming techniques (see e.g. Grossmann and Kravanja, 1997; Biegler and Grossmann, 2004). Recent developments in logic-based optimization (e.g. Grossmann and Biegler, 2004) are regarded as one of the most important achievements for effectively modeling and solving discrete-continuous synthesis problems. Although several general-purpose MINLP solvers (see www.gamsworld.org/minlp/solvers.html), including the logic-based solver LOGMIP (Vecchietti and Grossmann, 1997), 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. This paper reports on the experience gained in developing such a synthesis environment, and experiences gained when solving process network problems using up to several hundred discrete variables. Different formulations for logical interconnection nodes are applied and the following representations of outer approximations (OA) for the Outer Approximation/Equality Relaxation algorithm are compared:

Big-M formulation: $$h(x^l)+\nabla_x h(x^l)(x-x^{\mathrm{f}})\le M(1-y) \tag{1}$$

Convex hull representation: $$\nabla_x h(x^l)x \le \left(\nabla_x h(x^l)^{\mathrm{T}} x^l - h(x^l)\right)y \tag{2}$$

An alternative formulation: $$\nabla_x h(x^l)x \le \nabla_x h(x^l)x^{\mathrm{f}} + \left(\nabla_x h(x^l)^{\mathrm{T}}(x^l - x^{\mathrm{f}}) - h(x^l)\right)y \tag{3}$$

Unlike convex hull representation, where the continuous variables $\boldsymbol{x}$ are usually forced into zero values when the corresponding disjunctives are false, in the new formulation the variables are forced into arbitrarily-forced values, $\boldsymbol{x}^{\mathrm{f}}$.

We report our experience in the selection of different $\boldsymbol{x}^{\mathrm{f}}$ and implementation of different formulations in the MINLP process synthesizer MIPSYN (Mixed-Integer Process SYNthesizer), the successor of PROSYN-MINLP (Kravanja and Grossmann, 1994).

2. An alternative convex-hull representation

An efficient way of formulating discrete/continuous nonlinear problems in the area of process synthesis is to use Generalized disjunctive programming (GDP) (e.g. Türkay and Grossmann, 1996). One of the most important features of GDP is that NLPs are solved only in the reduced space of global and currently selected alternatives. The other important feature is that, before the first outer approximation disjunctive program (OADP) is solved, outer approximations (linearizations) are derived for the whole problem. Both features significantly improve efficiency when solving (OADP) problems. The conventional (OADP) is given in the following form:

$$
\begin{aligned}
&\min Z = \sum_k c_k + \alpha \\
&\text{s.t.} \quad \left.\begin{aligned} \alpha &\geq f(x^l) + \nabla_x f(x^l)^{\mathrm{T}}(x - x^l) \\ g(x^l) &+ \nabla_x g(x^l)^{\mathrm{T}}(x - x^l) \leq 0 \end{aligned}\right\}, \quad l = 1,\dots,L \\
&\qquad A^g(x) \leq b^g \\
&\begin{bmatrix} Y_{ik} \\ c_i = \gamma_i \\ x^{\mathrm{LO}} \leq x \leq x^{\mathrm{UP}} \\ A^{ik}(x) \leq b_{ik} \\ \nabla_x h_{ik}(x^l)^{\mathrm{T}} x \leq \nabla_x h_{ik}(x^l)^{\mathrm{T}} x^l - h_{ik}(x^l),\ l \in L_{ik} \end{bmatrix} \vee \begin{bmatrix} \neg Y_{ik} \\ c_i = 0 \\ B^{ik} x = 0 \\ \\ \end{bmatrix} \quad k \in SD, i \in D_k, \\
&\qquad \Omega(Y) = \text{true} \\
&x \in R^n, \quad c \in R^m, \quad Y \in \{true, false\}^m
\end{aligned}
\qquad \text{(OADP)}
$$

where qualitative logical and discrete decisions are represented by disjunctives ($i \in D_k$, $k \in SD$) and propositional logical constraints $\Omega(Y)$, whilst continuous quantitative decisions by (non)linear (in)equality constraints, which can be global ($g(x) \leq 0$, $A^g(x) \leq b^g$) or belong to local representations of alternatives ($h_{ik}(x) \leq 0$, $A^{ik}(x) \leq b^{ik}$). Note that when an alternative is not selected, its linearizations do not apply, and $\boldsymbol{x}$ is set to zero. Türkay and Grossmann (1996) developed convex-hull OAs for variables $\boldsymbol{x}$ that take zero or nonzero values by disaggregating vector $\boldsymbol{x}$ into sub vectors of zero $\boldsymbol{x}_{\mathrm{Z}}$ and nonzero $\boldsymbol{x}_{\mathrm{NZ}}$ variables. Here, an alternative and more general OADP is proposed, where vector $\boldsymbol{x}$ can be set to any value $\boldsymbol{x}^{\mathrm{f}}$ when the alternative is not selected:

$$
\begin{aligned}
&\min Z = \sum_k c_k + \alpha \\
&\text{s.t.} \quad \left.\begin{aligned} \alpha &\geq f(x^l) + \nabla_x f(x^l)^{\mathrm{T}}(x - x^l) \\ g(x^l) &+ \nabla_x g(x^l)^{\mathrm{T}}(x - x^l) \leq 0 \end{aligned}\right\}, \quad l = 1,\dots,L \\
&\qquad A^g(x) \leq b^g \\
&\begin{bmatrix} Y_{ik} \\ c_i = \gamma_i \\ x^{\mathrm{LO}} \leq x \leq x^{\mathrm{UP}} \\ A^{ik}(x) \leq b_{ik} \\ \nabla_x h_{ik}(x^l)^{\mathrm{T}} x \leq \nabla_x h_{ik}(x^l)^{\mathrm{T}} x^l - h_{ik}(x^l) \end{bmatrix} \vee \begin{bmatrix} \neg Y_{ik} \\ c_i = 0 \\ x = x^{\mathrm{f}} \\ A^{ik}(x - x^{\mathrm{LO}}) \leq b_{ik} \\ \nabla_x h_{ik}(x^l)^{\mathrm{T}} x \leq \nabla_x h_{ik}(x^l)^{\mathrm{T}} x^{\mathrm{f}} \end{bmatrix} \quad k \in SD, i \in I, l \in L_{ik} \\
&\qquad \Omega(Y) = \text{true} \\
&x \in R^n, \quad c \in R^m, \quad Y \in \{true, false\}^m
\end{aligned}
\qquad \text{(A-OADP)}
$$

Note that, auxiliary linear inequalities ($\nabla_x h_{ik}(x^l)^{\mathrm{T}} x \leq \nabla_x h_{ik}(x^l)^{\mathrm{T}} x^{\mathrm{f}}$) are applied in order to preserve the feasibility of OAs in MILP when an alternative is not selected and the corresponding $\boldsymbol{x}$ is set to $\boldsymbol{x}^{\mathrm{f}}$. By replacing Y_{ik} in (A-OADP) with binary variable y_{ik}, the following alternative convex-hull formulation for OAs can be derived at:

$$\nabla_x h_{ik}\left(x^l\right)^{\mathrm{T}} x \le \left(\nabla_x h_{ik}\left(x^l\right)^{\mathrm{T}} x^l - h_{ik}\left(x^l\right)\right) y_{ik} + \nabla_x h_{ik}\left(x^l\right)^{\mathrm{T}} x^{\mathrm{f}} (1 - y_{ik}) \tag{4}$$

which can finally take the form:

$$\nabla_x h_{ik}\left(x^l\right)^{\mathrm{T}} x \le \nabla_x h_{ik}\left(x^l\right)^{\mathrm{T}} x^{\mathrm{f}} + \left(\nabla_x h_{ik}\left(x^l\right)^{\mathrm{T}} (x^l - x^{\mathrm{f}}) - h_{ik}\left(x^l\right)\right) y_{ik} \tag{5}$$

In addition, in order to set $\boldsymbol{x}$ to $\boldsymbol{x}^{\mathrm{f}}$ when an alternative is not selected, the following constraints should be applied:

$$x \le x^{\mathrm{f}} + (x^{\mathrm{UP}} - x^{\mathrm{f}}) y_{ik} \tag{6}$$

$$x \ge x^{\mathrm{f}} + (x^{\mathrm{LO}} - x^{\mathrm{f}}) y_{ik} \tag{7}$$

The key feature of the alternative OAs (eq. 5) is that they preserve feasibility, even in the presence of nonconvexities when alternatives are not selected and $\boldsymbol{x}$ is set to $\boldsymbol{x}^{\mathrm{f}}$. This enables the use of variables with nonzero lower bounds, directly without additional logical constraints on the variables. Note that when $\boldsymbol{x}^{\mathrm{f}}$ is equal to the lower bounds ($\boldsymbol{x}^{\mathrm{f}} = \boldsymbol{x}^{\mathrm{LO}}$), inequality (7) becomes redundant and can be omitted from the formulation. Similarly, ineq. (6) can be omitted when $\boldsymbol{x}^{\mathrm{f}}$ is equal to $\boldsymbol{x}^{\mathrm{UP}}$. This reduces the size of the MILP problem. An interesting feature of the proposed formulation of OAs (ineq. 5) is that nonzero $\boldsymbol{x}^{\mathrm{f}}$ can be chosen, such that linearization coefficients at $\boldsymbol{y}$ become zero, and the mixed-integer OAs become pure-continuous constraints that are much easier to solve, especially when the number of binary variables is very high. However, forcing $\boldsymbol{x}$ to a nonzero $\boldsymbol{x}^{\mathrm{f}}$, transforms pure-continuous linear constraints $A^{ik}\left(x\right) \le b_{ik}$ into mixed-integer constraints $A^{ik}\left(x - x^{\mathrm{LO}} y_{ik}\right) \le b_{ik}$. It is then obvious that the selection of $\boldsymbol{x}^{\mathrm{f}}$ and, especially the selection of the most suitable OA and modeling representation, may not be a straightforward task and may significantly influence the efficiency of the search. The earliest experience indicates that the best efficiency is achieved when $\boldsymbol{x}^{\mathrm{f}}$ is set to $\boldsymbol{x}^{\mathrm{LO}}$. A procedure for a systematic selection of the most suitable $\boldsymbol{x}^{\mathrm{f}}$ is under way. Until recently only big-M formulation of OAs and big-M representation of logical interconnection nodes (single-choice mixers and splitters) were used in MIPSYN to solve MINLP synthesis problems. Now, OAs and logical interconnected nodes are also represented by the conventional convex-hull and the alternative convex-hull formulations.

3. Examples

Three synthesis problems of different sizes and complexities are solved using all three OAs and modeling representations, in order to test and compare their efficiencies. The first numerical example is a network synthesis example with a simple model but very large-scale combinatorics with 400 binary variables. The second example is the synthesis of heat exchanger network (HEN) comprising different types of exchangers. The model exhibits moderate complexity and high combinatorics (249 binary variables). The last, alil chloride example, is the synthesis of a reactor/separator network in to an overall heat integrated process scheme, with a complex model and smaller-size combinatorics (32 binary variables).

3.1. Network synthesis problem

Fig. 1 shows a superstructure comprising a sequence of exclusive-or alternatives. This model consists of a linear objective function, nonlinear design equations, formulation for single-choice splitters and mixers and exclusive-or logical constraints (detailed formulations will be given in the extended paper). The objective is to minimize total cost at the fixed demand of the final outflow. The problem was solved by using all three OA and modeling representations. Solution statistics until the 3rd major MINLP iteration is reported in Table 1.

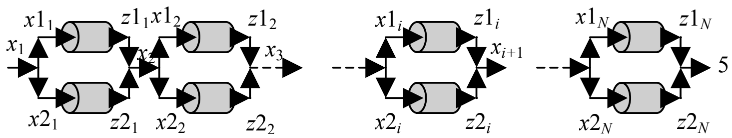

Figure 1: Superstructure of the network synthesis problem.

Table 1: Solution statistics of the network synthesis problem.

	Best NLP	**Integrality gap, %**	**No. of eq./ No.of var.**	**No.of nodes**	**CPU for 3 it., sec.**	**Nodes/CPU for 3 it.**
BigM	n/a	n/a	3802/1801	n/a	n/a	n/a
Convex-hull	183.870	0.868	3402/1801	319	15.071	21.2
ACH ($\boldsymbol{x}^{\mathrm{f}} = \boldsymbol{x}^{\mathrm{LO}}$)	183.870	0.868	2202/1801	293	4.274	68.6
ACH ($\boldsymbol{x}^{\mathrm{f}} = \boldsymbol{x}^{\mathrm{UP}}$)	183.870	0.868	3402/1801	2264	46.209	49.0
ACH ($\boldsymbol{x}^{\mathrm{f}} = \boldsymbol{x}^{1}$)	183.870	0.868	3402/1801	341	24.142	14.1

As can be seen in Table 1, it was impossible with big-M formulation to solve the problem in a reasonable time, whilst both convex-hull representations enable the solving of this high combinatorial problem very quickly. Also it can be seen that for the alternative convex-hull formulation (ACH) the selection of $\boldsymbol{x}^{\mathrm{f}}$ is very important and that the best efficiency of the search is achieved when $\boldsymbol{x}^{\mathrm{f}} = \boldsymbol{x}^{\mathrm{LO}}$. Note that with the same integrality gap and somewhat smaller number of constraints, the alternative formulation with $\boldsymbol{x}^{\mathrm{f}} = \boldsymbol{x}^{\mathrm{LO}}$ could solve the problem in only a quarter of the CPU time needed to solve the problem using the conventional convex-hull formulation.

3.2. HEN synthesis problem

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 when the match is rejected. The model is described in detail by Soršak and Kravanja (2002).
Consideration of different types of exchangers enables the simultaneous selection of exchanger types; however, it significantly increases the number of binary variables. In this example of 4 hot and 5 cold process streams and 4 stages, the problem originally had 320 binary variables. By prescreening alternatives the number was reduced to 249. Table 2 shows statistics for three different representations. With respect to integrality gap, CPU time and the number of nodes, both convex-hull representations outperform the big-M one whilst the efficiency of the alternative convex-hull formulation is slightly better than the conventional formulation one. Also, with big-M, a slightly inferior solution was obtained than with the convex-hull representations.

Table 2: Solution statistics for the HEN synthesis problem.

	Best NLP	**Integrality gap, %**	**No. of eq./ No.of var.**	**No.of nodes**	**CPU for 8 it., sec.**	**Nodes/CPU for 8 it.**
BigM	821.00	31.321	8414/5595	18950	86.050	220.2
Convex-hull	818.69	7.465	6814/5595	4817	29.779	161.8
ACH ($\boldsymbol{x}^{\mathrm{f}} = \boldsymbol{x}^{\mathrm{LO}}$)	818.69	7.465	5534/5595	4065	28.207	144.0

3.3. Alil chloride example

Details of the alil chloride problem are given by Iršič-Bedenik et al. (2004). The reactor/separator superstructure comprises a series of basic reactor substructure elements with side streams and intermediate separators at different locations. In each element a recycle reactor (a recycle stream around a PFR) and a CSTR are embedded in parallel arrangement so as to enable a different feeding, recycling and bypassing. In addition, 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 a multi-utility configuration model (Duran and Grossmann, 1986). The overall model is highly nonlinear and nonconvex. 32 binary variables were assigned to discrete decisions. The objective is to maximize the net present value at a fixed production for alil chloride. The solution statistics of all three OA and modeling representations is given in Table 3.

Table 3: Solution statistics of alil chloride problem.

	Best NLP k$/a	**Integrality gap, %**	**No. of eq./ No.of var.**	**No.of nodes**	**CPU for 7 it., sec.**	**Nodes/CPU for 7 it.**
BigM OAs	83.709	0.348	2046/10426	568	66.027	8.6
Convex-hull	83.679	0	4408/10426	53	10.567	5.0
ACH ($\boldsymbol{x}^{\mathrm{f}} = \boldsymbol{x}^{\mathrm{LO}}$)	86.245	0	3903/10426	9	5.866	1.5

When logical constraints ($\boldsymbol{x} \leq \boldsymbol{x}^{\mathrm{UP}}\boldsymbol{y}$) are imposed on all continuous variables presented as alternatives, integrality gaps of both convex-hull approaches are decreased practically to zero which significantly facilitates the efficiencies of the first couple of MILPs. However, in the tighter MILP representations, the effects of nonconvexities become more severe, causing a significant increase in the number of nodes in the subsequent MILPs. It is interesting to note that, due to the presence of nonconvexities, even with the zero integrality gaps the efficiencies of the convex-hull representations do not improve. In order to decrease the troublesome effect of nonconvexities a special convex test (Kravanja and Grossmann, 1994) was applied and violating OAs were temporarily dropped out of the master MILPs. Statistics of solutions for both convex-hull representations are now significantly improved (Table 3), especially in the case of alternative convex-hull representation, where the best solution was found and the least computational effort was needed to obtain it.

4. Conclusion

The main aim of this research is oriented towards the development of an advanced and robust synthesizer shell, capable of solving large-scale applications in different engineering domains. The performances of different OA and modeling representations are summarized in Table 4. Both convex-hull representations usually outperform the big-M one. The earliest high performance solutions with alternative representation, indicates that the alternative convex-hull representation could be more efficient in solving high combinatorial problems than the conventional one and has the smallest problem size. On the other side it exhibits the strongest sensitivity to the effects of nonconvexities and the model formulation is probably the most complicated. It should be noted that so far the research has been focused only to the OA algorithm. The application of the alternative convex-hull formulation with other MINLP techniques is under way.

Table 4: Performance of different OA and modeling representations.

	Big-M	**Convex-hull**	**Alternative $x^{\mathrm{f}} = x^{\mathrm{LO}}$**
Easiness of modeling	The most easy	Moderate	The most complicated
Problem size	From the smallest to the largest	The largest	Moderate
Effect of nonconvexities	The smallest	Moderate	The strongest
Nodes/sec of CPU time	The largest	Moderate	The smallest or moderate

References

Biegler L.T. and I. E. Grossmann (2004). Retrospective on optimization, Computers and Chemical Engineering, 28, 1169-1192.

Duran, M.A. and I.E. Grossmann (1986). Simultaneous optimization and heat integration of chemical processes, AIChE J. 32, 123-138.

Grossmann, I. E. and Z. Kravanja, Mixed-integer nonlinear programming: a survey of algorithms and applications. V: Biegler, L. T. (ed.). Large-scale optimization with applications. Part 2, Optimal design and control, (The IMA volumes in mathematics and its applications, Vol. 93). New York: Springer-Verlag, 73-100 (1997).

Grossmann, I. E. and L. T. Biegler (2004). Part II. Future perspective on optimization, Computers and Chemical Engineering, 28, 1193-1218.

Iršič Bedenik N., B. Pahor and Z. Kravanja (2004). An integrated strategy for the hierarchical multilevel MINLP synthesis of overall process flowsheets using the combined synthesis/analysis approach, Computers and Chemical Engineering, 28, 693-706.

Kravanja Z. and I.E. Grossmann (1994). New Developments and Capabilities in PROSYN – an Automated Topology and Parameter Synthesizer, Computers and Chemical Engineering, 18, 1097-1114.

Soršak A. and Z. Kravanja (2002). Simultaneous MINLP synthesis of heat exchanger networks comprising different exchanger types. Computers and Chemical Engineering, 26, 599-615.

Türkay M. and I. E. Grossmann (1996). A logic based outer-aproximation algorithm for MINLP optimization of process flowsheets. Computers and Chemical Engineering, 20, 959-978.

Vecchietti A. and I.E. Grossmann (1997). LOGMIP: a disjunctive 0-1 nonlinear optimizer for process systems models, Computers and Chemical Engineering, 21, 427-432.

www.gamsworld.org/minlp/solvers.html, GAMS MINLP solvers (2005).