18<sup>th</sup> European Symposium on Computer Aided Process Engineering – ESCAPE 18 Bertrand Braunschweig and Xavier Joulia (Editors) © 2008 Elsevier B.V./Ltd. All rights reserved.

# Modeling Piecewise Under- and Overestimators for Bilinear Process Network Synthesis via Mixedinteger Linear Programming

Danan Suryo Wicaksono and I. A. Karimi\*

Department of Chemical and Biomolecular Engineering, National University of Singapore, 4 Engineering Drive 4, Singapore 117576, Singapore \*corresponding author; E-mail: cheiak@nus.edu.sg

### Abstract

Piecewise under- and overestimators that rely on mixed-integer linear programming (MILP) have been recently proposed for the global optimization of bilinear program (BLP), which is essential for process network synthesis. Modeling is an important aspect for piecewise MILP under- and overestimators. Here, we propose a novel class of models, called as hybrid models. We also extend the modeling approach from univariate partitioning to bivariate partitioning. A case study using a benchmark process network synthesis problem shows several interesting results.

Keywords: process network synthesis, piecewise under- and overestimators

### **1. Introduction**

Process network synthesis deals with the generation of alternative flowsheets and the selection of an alternative whose configuration and parameters best satisfy a given criterion which is commonly economic and / or technical in nature. One effective tool for process network synthesis is the so-called *superstructure* [1] where a number of process design alternatives are represented as a network of interconnected arcs and nodes. Later, this graphical representation is transformed into a mathematical program. In many cases, the resulting mathematical program contains nonconvexity in order to model the complex interaction within the process. This property makes the mathematical program not only hard to solve to global optimality, but also difficult to even locate a feasible solution.

One of the most recurring nonconvex terms in process network synthesis problems is the bilinear term. A bilinear term comes from the product of two continuous variables from different variable sets (e.g. flow rates and compositions) and becomes linear when one of the variables involved is fixed. *Bilinear process network synthesis* is a process network synthesis problem in which the nonlinear terms arise solely from the bilinear terms and thus is a BLP. The common approach [2] to obtain the global optimal solution of this nonconvex program requires that every occurrence of the bilinear term *xy* over the rectangle  $x^L \le x \le x^U$ ,  $y^L \le y \le y^U$  is substituted by an additional variable *z* bounded by convex underestimators [Eq. (A-1) - (A-2)] and concave overestimators [Eq. (A-3) - (A-4)] representing the convex envelope [3, 4]. The resulting problem is a linear program (LP), called as *LP relaxation*, whose solution provides a lower (upper) bound for the global minimum (maximum) solution.

$$z \ge x^{L} \cdot y + x \cdot y^{L} - x^{L} \cdot y^{L} \qquad (A-1) \qquad z \ge x^{U} \cdot y + x \cdot y^{U} - x^{U} \cdot y^{U} \qquad (A-2)$$

$$z \le x^{L} \cdot y + x \cdot y^{U} - x^{L} \cdot y^{U} \quad (A-3) \qquad z \le x^{U} \cdot y + x \cdot y^{L} - x^{U} \cdot y^{L} \quad (A-4)$$

A recent advance has been to utilize an improved version of this convex envelope via ab *initio* partitioning of the search space using binary variables, called as *piecewise MILP* relaxation. Such a relaxation has been shown to be able to improve the relaxation quality and computational efficiency of certain global optimization algorithms in solving several process network synthesis problems [5-7]. One key issue is the modeling aspect of the piecewise MILP under- and overestimators which determines the computational effort required to solve the piecewise MILP relaxation problem. This aspect is crucial since the piecewise MILP relaxation is typically solved repeatedly and is reported as the most time consuming step. Wicaksono and Karimi, 2007 [8] developed several novel competitive models and compared them with existing models using several case studies. They further studied the properties of piecewise MILP underand overestimators within a two-level-relaxation framework as well as presented several important insights and theoretical results. However, all the existing works [5-8] have focused on partitioning the search space on the domain of one variable set (either x or y), called as *univariate partitioning*. *Bivariate partitioning*, which partitions the search space on both variable sets (both x and y), has not been addressed so far and offers advantages as shown and discussed later.

In this work, we first propose another class of piecewise MILP under- and overestimators which combine the attractive properties of models presented by Wicaksono and Karimi, 2007 [8]. Furthermore, we propose several novel models for bivariate partitioning. We compare the computational performance of these models using a benchmark case study of non-sharp distillation column sequencing. We analyze the results using two-level-relaxation framework introduced by Wicaksono and Karimi, 2007 [8].

# 2. Univariate Partitioning

In univariate partitioning, only the search space of one set of variable x involved in bilinear terms xy is divided into M segments through the introduction of grid points  $a(m) [x^L = a(1) < ... < a(M + 1) = x^U; d(m) = a(m + 1) - a(m)]$  while the search space of other set y remains intact. Previous works have used univariate partitioning exclusively in their works [5-8]. Wicaksono and Karimi, 2007 [8] developed, compared, and studied various models for univariate partitioning. Their modeling approaches can be classified into three classes: *big-M*, *convex combination*, and *incremental cost*. They showed that big-M models exhibit poor 2<sup>nd</sup> level relaxation quality and thus are likely not competitive for piecewise MILP under- and overestimators due to the nature of the problem in which the feasible region of each disjunct is disjoint to each other. On the other hand, convex combination and incremental cost models offer better 2<sup>nd</sup> level relaxation quality which represent the convex hull.

#### 2.1. Big-M and Hybrid Models

In this work, we propose a novel model which is a hybrid model. This novel hybrid model **H-1** does not require additional continuous variables as in a pure big-M model **B-1**. However, it exhibits tighter  $2^{nd}$  level relaxation quality resembling those representing the convex hull. **H-1** is obtained by augmenting **B-1** [Eq. (B-1) - (B-8)] with the continuous convex envelope [Eq. (A-1) - (A-4)] over the rectangle  $x^{L} \le x \le x^{U}$ ,  $y^{L} \le y \le y^{U}$ . *BM* is a sufficiently large number required. In this sense, big-M constraints convey the disjunctive logic of partitioning the search space while the convex envelope over the entire region promotes the  $2^{nd}$  level relaxation quality. We argue that this model has a relation with the *lift-and-project* method proposed by Balas, 1998 [9]. However, the cuts describing the hull are not obtained via expensive *lift-and-project*. Inspiration for such a formulation is drawn from the results of Wicaksono and Karimi, 2007 [8].

$$\lambda_{x}(m) = \begin{cases} 1 & \text{if } a(m) \le x \le a(m+1) \\ 0 & \text{otherwise} \end{cases} \qquad \qquad x \ge a(m) \cdot \lambda_{x}(m) + x^{L} \cdot [1 - \lambda_{x}(m)] \\ \forall m \quad (B-1) \end{cases} \qquad \qquad \qquad \forall m \quad (B-3)$$

$$\sum_{m=1}^{M} \lambda_x(m) = 1$$
(B-2)
$$x \le a(m+1) \cdot \lambda_x(m) + x^U \cdot [1 - \lambda_x(m)]$$

$$\forall m \quad (B-4)$$

$$z \ge x \cdot y^{L} + a(m) \cdot y - a(m) \cdot y^{L} - BM \cdot [1 - \lambda_{x}(m)] \qquad \forall m \qquad (B-5)$$

$$z \ge x \cdot y^U + a(m+1) \cdot y - a(m+1) \cdot y^U - BM \cdot [1 - \lambda_x(m)] \quad \forall m$$
(B-6)

$$z \le x \cdot y^{U} + a(m) \cdot y - a(m) \cdot y^{U} + BM \cdot [1 - \lambda_{x}(m)] \qquad \forall m \qquad (B-7)$$

$$z \le x \cdot y^{L} + a(m+1) \cdot y - a(m+1) \cdot y^{L} + BM \cdot [1 - \lambda_{x}(m)] \quad \forall m$$
(B-8)

#### 2.2. Incremental Cost Model

In order to compare the computational performance of the proposed hybrid model, we use one of the models from Wicaksono and Karimi, 2007 [8]. This model **I-1** belongs to the class of incremental cost models and comprises of Eq. (C-1) - (C-9). **I-1** requires additional non-negative continuous variables  $\Delta u(m)$  and  $\Delta w_x(m)$ .

$$\theta_x(m) = \begin{cases} 1 & \text{if } x \ge a(m+1) \\ 0 & \text{otherwise} \end{cases} \forall m \text{ (C-1)} \qquad x = x^L + \sum_{m=1}^M d_x(m) \cdot \Delta u(m) \quad \text{(C-2)} \end{cases}$$

$$z = x^{L} \cdot y + y^{L} \cdot x - x^{L} \cdot y^{L} + \sum_{m=1}^{M} d_{x}(m) \cdot \Delta w_{x}(m)$$
(C-3)

$$0 \le \Delta u(M) \le \theta_x(M-1) \qquad (C-4) \qquad \theta_x(1) \le \Delta u(1) \le 1 \qquad (C-5)$$

$$\theta_x(m) \le \Delta u(m) \le \theta_x(m-1)$$
  $\forall m \in [2, M-1]$  (C-6)

$$\Delta w_x(m) \ge (y^U - y^L) \cdot \Delta u(m) + y - y^U \qquad \forall m \qquad (C-7)$$

$$\Delta w_{x}(m) \leq (y^{U} - y^{L}) \cdot \Delta u(m) \qquad \forall m \qquad (C-8)$$

$$\Delta w_x(m) \le y - y^L \qquad \qquad \forall m \qquad (C-9)$$

## 3. Bivariate Partitioning

Unlike the univariate partitioning, bivariate partitioning divides the search space of both sets of variables *x* and *y* respectively into *M* and *N* segments via the introduction of grid points a(m) and  $b(n) [x^L = a(1) < ... < a(M + 1) = x^U; d_x(n) = a(m + 1) - a(m)$  and  $y^L = b(1) < ... < b(N + 1) = y^U; d_y(n) = b(n + 1) - b(n)]$ . To our knowledge, such a model has not been reported in the literature. Here, we introduce such a model using the newly

proposed hybrid approach and extend the incremental cost model already proposed by Wicaksono and Karimi, 2007 [8] to use bivariate partitioning.

### 3.1. Big-M and Hybrid Models

Extension of the hybrid model to bivariate partitioning **H-2** is obtained by introducing another set of binary variables representing segments in *y*. **H-2** contains Eq. (A-1) - (A-4), (B-1) - (B-4), and (D-1) - (D-8). Without Eq. (A-1) - (A-4), **H-2** reduces to **B-2**.

$$\lambda_{y}(n) = \begin{cases} 1 & \text{if } b(n) \le y \le b(n+1) \\ 0 & \text{otherwise} \end{cases} \qquad y \ge b(n) \cdot \lambda_{y}(n) + y^{L} \cdot [1 - \lambda_{y}(n)] \\ \forall n \quad (D-3) \end{cases}$$

$$\forall n$$
 (D-1)

$$\sum_{n=1}^{N} \lambda_{y}(n) = 1 \qquad (D-2) \qquad y \leq b(n+1) \cdot \lambda_{y}(n) + y^{U} \cdot [1 - \lambda_{y}(n)] \\ \forall n \quad (D-4)$$

$$z \ge x \cdot b(n) + a(m) \cdot y - a(m) \cdot b(n) - BM \cdot [2 - \lambda_x(m) - \lambda_y(n)] \quad \forall [m, n] \quad (D-5)$$
$$z \ge x \cdot b(n+1) + a(m+1) \cdot y - a(m+1) \cdot b(n+1) - BM \cdot [2 - \lambda_x(m) - \lambda_y(n)]$$

$$\forall [m,n] \quad (D-6)$$

$$z \le x \cdot b(n+1) + a(m) \cdot y - a(m) \cdot b(n+1) + BM \cdot [2 - \lambda_x(m) - \lambda_y(n)]$$

$$\forall [m,n] \quad (D-7)$$

$$z \le x \cdot b(n) + a(m+1) \cdot y - a(m+1) \cdot b(n) + BM \cdot [2 - \lambda_x(m) - \lambda_y(n)]$$

$$\forall [m,n] \quad (D-8)$$

#### 3.2. Incremental Cost Model

The extension of incremental cost model to bivariate partitioning **I-2** is obtained in a similar fashion. **I-2** contains Eq. (C-1) - (C-2), (C-4) - (C-6) and (E-1) - (E-9). **I-2** requires additional non-negative continuous variables  $\Delta u(m)$ ,  $\Delta v(n)$ , and  $\Delta w_{xy}(m,n)$ .

$$\theta_{y}(n) = \begin{cases} 1 & \text{if } y \ge b(n+1) \\ 0 & \text{otherwise} \end{cases} \forall n \text{ (E-1)} \qquad y = y^{L} + \sum_{n=1}^{N} d_{y}(n) \cdot \Delta v(n) \quad (E-2) \end{cases}$$

$$z = x^{L} \cdot y + y^{L} \cdot x - x^{L} \cdot y^{L} + \sum_{m=1}^{M} \sum_{n=1}^{N} d_{x}(m) \cdot d_{y}(n) \cdot \Delta w_{xy}(m,n)$$
(E-3)

$$0 \le \Delta v(N) \le \theta_y(N-1)$$
 (E-4)  $\theta_y(1) \le \Delta v(1) \le 1$  (E-5)

$$\theta_{y}(n) \le \Delta v(n) \le \theta_{y}(n-1)$$
  $\forall n \in [2, N-1]$  (E-6)

$$\Delta w_{xy}(m,n) \ge \Delta u(m) + \Delta v(n) - 1 \qquad \forall [m,n] \qquad (E-7)$$

$$\Delta w_{xy}(m,n) \le \Delta u(m) \ \forall [m,n] \ (E-8) \qquad \Delta w_{xy}(m,n) \le \Delta v(n) \ \forall [m,n] \ (E-9)$$

18<sup>th</sup> European Symposium on Computer Aided Process Engineering – ESCAPE 18 Bertrand Braunschweig and Xavier Joulia (Editors) © 2008 Elsevier B.V./Ltd. All rights reserved.

## 4. Case Study

Numerical study provides insight into the actual computational performance of the proposed models. In this study, we use a benchmark process network synthesis problem from Floudas *et al.*, 1999 [10]. It is a non-sharp distillation column sequencing problem which was formulated in the form of a BLP with 24 variables, 18 constraints, and 12 bilinear terms. In addition, the upper bounds on all flow rate variables are contracted to 180 Kgmol/h and the lower bounds for the flow rate of stream 18 is contracted to 10 Kgmol/h without cutting off the reported global optimal solution.

The six models mentioned before (i.e. **B-1**, **H-1**, **I-1**, **B-2**, **H-2**, **I-2**) were run to solve the problem in the platform of Dell Precision PW690 workstation with 3 GHz Intel Xeon single processor, 64 GB RAM, and Windows XP Professional x64 as operating system. LP and MILP problems were solved using GAMS 22.2 / CPLEX 10. Relative gap tolerance was set to zero in order to ensure solution optimality. The value of *BM* is set to  $(x^U - x^L) (y^U - y^L)$ . Different sets of grid points are generated using the formula presented in Wicaksono and Karimi, 2007 [8] with various values of  $\alpha$  and  $\beta$  for variables set *x* and *y*, respectively. In all cases involving bivariate partitioning,  $\alpha$  is set to be equal to  $\beta$ . *Piecewise Gain* (PG) and *Relaxed Piecewise Gain* (RPG) respectively compare the bounds on the global optimal solution obtained from the 1<sup>st</sup> and 2<sup>nd</sup> level piecewise MILP relaxations versus that from LP relaxation [8].

Bivariate partitioning was done with 12 segments on each flow rate and composition. Such a number was chosen since it gives an interesting result regarding global optimality as discussed later and provides easier benchmarking. For a fair comparison, univariate partitioning was done with 20 segments on flow rate and 30 segments on composition. The additional segments are used to compensate for the absence of simultaneous partitioning in flow rate and composition involved in bilinear terms. Thus, all case studies use the same number of total segments.

#### 5. Results and Discussion

In all cases tested (see Table 1 and Table 2), the pure big-M perfoms much worse than other models in terms of computational time. It is clear that the novel hybrid model is more competitive than the pure big-M model due to the enhanced 2<sup>nd</sup> level relaxation quality. Although quite competitive with an instance of incremental cost models in several examples, the hybrid model does not perform faster in bivariate partitioning case tested. This issue requires further study. The incremental cost model requires one less binary variable to model the same number of segments in one domain as compared to other models. Consequently, the use of bivariate partitioning reduces further the number of binary variables in incremental cost models. Note that in this study, incremental cost models require 114, 116, and 110 binary variables for univariate partitioning on flow rate, univariate partitioning on composition, and bivariate partitioning, respectively. Other models require 120 binary variables for both univariate and bivariate partitioning. For all cases, it was found that  $PG \ge 0$  and  $RPG \le 0$ . Moreover, RPG < 0 for big-M models, RPG = 0 for incremental cost and hybrid models, and PG is independent of model. All these results are in line with those of Wicaksono and Karimi, 2007 [8] suggesting the generality of such results as they apply not only for univariate partitioning but also for bivariate partitioning. Results in this study show that the bivariate partitioning give higher PG than univariate partitioning for equivalent numbers of segments and grid points positioning. In fact, the 1<sup>st</sup> level optimal solution obtained via bivariate partitioning for  $\alpha = \beta = 1$  is equal to the global optimal solution reported by

Floudas *et al.*, 1999 [10]. Thus, the corresponding PG represents the maximum attainable PG ( $PG_{max}$ ) for this case, i.e. 0.458. Obviously, the advantage of obtaining higher PG is balanced by the increase of computational cost. Thus, an efficient use of such a method requires more study.

	I								
model	α	PG	RPG	CPU s	model	α	PG	RPG	CPU s
flow rate $M = 20$					composition M = 30				
B-1	1.00	0.205	-0.218	1.453	B-1	1.00	0.394	-0.220	3.875
	1.50	0.296	-0.219	5.624		1.50	0.374	-0.220	2.125
H-1	1.00	0.205	0	0.406	H-1	1.00	0.394	0	0.656
	1.50	0.296	0	0.828		1.50	0.374	0	0.312
I-1	1.00	0.205	0	0.765	I-1	1.00	0.394	0	1.203
	1.50	0.296	0	0.749		1.50	0.374	0	0.765
Table 2. Computational results for bivariate partitioning $(M = N = 12)$									
model		<b>α =</b> β		PG		RPG		CPU s	
B-2	1.00		0.458		-0.220		6660.176		
	1.50		0.443		-0.220		5670.135		
H-2	1.00			0.458		0		2019.112	
		1.50		0.443		0		3339.869	)
I-2	1.00		0.458		0		17.812		
		1.50		0.443		0		11.234	

Table 1. Computational results for univariate partitioning

# 6. Conclusion

The novel hybrid model, without any additional continuous variables, exhibits stronger  $2^{nd}$  level relaxation and performs faster than the big-M model, although it does not outperform one instance of the incremental cost model in the reported case study. The case study results suggest that the bivariate partitioning can provide better  $1^{st}$  level relaxation quality than the univariate partitioning. Some properties on *PG* and *RPG* hold for both the univariate and bivariate partitioning.

#### References

[1] L.T. Biegler, et al., Systematic Methods of Chemical Process Design, U.S.A., 1997.

- [2] R. Horst and H. Tuy, Global Optimization : Deterministic Approaches (2<sup>nd</sup> Ed.), Germany, 1993.
- [3] G.P. McCormick, Math. Prog., 10 (1976) 146.
- [4] F.A. Al-Khayyal and J.E. Falk, Math. Oper. Res., 8 (1983) 273.
- [5] R. Karuppiah and I.E. Grossmann, Comput. Chem. Eng., 30 (2006) 650.
- [6] M.L. Bergamini, *et al.*, Comput. Chem. Eng., 29 (2005) 1914.
- [7] C.A. Meyer and C.A. Floudas, AIChE J., 52 (2006) 1027.
- [8] D.S. Wicaksono and I.A. Karimi, AIChE J., (2007) accepted for publication.
- [9] E. Balas, Disc. Appl. Math., 89 (1998) 3.
- [10] C.A. Floudas, et al., Handbook of Test Problems in Local and Global Optimization, The Netherlands, 1999.