AIChE

Piecewise Linear Relaxation of Bilinear Programs Using Bivariate Partitioning

M. M. Faruque Hasan and I.A. Karimi
Dept. of Chemical and Biomolecular Engineering, National University of Singapore,
4 Engineering Drive 4, Singapore 117576, Singapore

DOI 10.1002/aic.12109
Published online November 30, 2009 in Wiley InterScience (www.interscience.wiley.com).

Several operational and synthesis problems of practical interest involve bilinear terms. Commercial global solvers such as BARON appear ineffective at solving some of these problems. Although recent literature has shown the potential of piecewise linear relaxation via ab initio partitioning of variables for such problems, several issues such as how many and which variables to partition, which partitioning scheme(s) and relaxation model(s) to use, placement of grid points, etc., need detailed investigation. To this end, we present a detailed numerical comparison of univariate and bivariate partitioning schemes. We compare several models for the two schemes based on different formulations such as incremental cost (IC), convex combination (CC), and special ordered sets (SOS). Our evaluation using four process synthesis problems shows a formulation using SOS1 variables to perform the best for both partitioning schemes. It also points to the potential usefulness of a 2-segment bivariate partitioning scheme for the global optimization of bilinear programs. We also prove some simple results on the number and selection of partitioned variables and the advantage of uniform placement of grid points (identical segment lengths for partitioning). © 2009 American Institute of Chemical Engineers *AIChE J,* 56: 1880–1893, 2010
Keywords: global optimization, bilinear program, piecewise under- and overestimators, process network synthesis, bivariate partitioning

Introduction

The mass and energy balance equations in many chemical engineering problems of practical interest (e.g., synthesis of process/energy/water networks,[1–5] pooling problems,[1,6,7] scheduling of crude oil and refinery blending operations,[8–10] distillation column sequencing,[11] and fuel networks[12] for oil and gas processing industries) often involve products of temperature and flow rate, enthalpy and flow rate, flow rate and quality, flow rate and composition, etc. When such equations appear in an optimization formulation, and both components (flow rate, temperature, enthalpy, etc.) of a product term are decision variables, then we have a bilinear term. Continuous optimization problems with at least one bilinear term and everything else being linear are called bilinear programs (BLPs). A mixed-integer bilinear program (MIBLP) is a BLP in which some decision variables are binary. Discrete structural and/or operational decisions (e.g., selecting treatment technologies for wastewater network or tanks for loading, unloading, or blending) result in such binary variables in a MIBLP.

BLP (and consequently MIBLP) is a nonconvex[13] optimization problem, so local NLP solvers such as CONOPT, MINOS, SNOPT, MSNLP, LGO cannot guarantee a globally optimal solution. Even global solvers such as BARON fail to converge or give a feasible solution to many such problems[9–10] of practical interest. Therefore, attaining globally optimal solutions for nonconvex BLPs and MIBLPs is a real, important, and challenging issue.

Recently, Pham et al.[7] proposed a heuristic approach for obtaining near-global solutions to pooling problems by discretizing pooling qualities to eliminate the bilinear terms. The resulting MILP is solved repeatedly with progressively finer discretizations to obtain the desired solution accuracy.

Correspondence concerning this article should be addressed to I.A. Karimi at cheiak@nus.edu.sg.

© 2009 American Institute of Chemical Engineers

Although their approach uses many binary variables, it solves some benchmark problems to near-global optimality much faster than some global solvers.

Many deterministic global optimization (GO)[13–16] techniques used for solving BLPs use the spatial branch-and-bound (sBB) algorithm.[17,18] The performance of such algorithms depends critically on the branching strategy and quality of solution bounds among others. Convex relaxation techniques, which are commonly used to obtain these solution bounds, can consume a significant portion of the computation time at each node.[2] Furthermore, poor relaxations can give loose bounds and slow down such algorithms considerably. Thus, both relaxation quality and efficient solution of relaxed subproblems are critical.

A common relaxation strategy for nonconvex factorable BLPs is to replace each bilinear term by its convex envelopes.[19,20] Several GO frameworks[21–24] have used this strategy called linear programming (LP) relaxation. Although the LP relaxation offers simplicity and solution efficiency, its relaxation quality (and thus solution bounds) can be poor.

Another relaxation technique[1,2,25–27] used an ab initio partitioning of the search domain into multiple smaller subdomains with separate LP relaxations. This piecewise linear relaxation of bilinear terms has attracted interest in process synthesis,[25] generalized pooling,[1] heat exchanger networks,[28] and integrated water use and treatment.[2] The need to combine the individual relaxations in a seamless manner gives rise to a MILP. Wicaksono and Karimi[26] developed and compared several MILP formulations for obtaining such piecewise linear relaxation. These MILP formulations partition the domains of selected variables appearing in the bilinear terms into exclusive and exhaustive segments. Thus, the choices of variables to partition and segment lengths are the key issues. Because a bilinear term has two variables, three possible choices for partitioning are obvious. Two of these involve partitioning only one of the two variables. This can be termed[26,27] univariate partitioning. The third choice is to partition both the variables. This can be called bivariate[27] partitioning.

All previously reported formulations for piecewise linear relaxation (MILP relaxation), except the preliminary work of Wicaksono and Karimi,[27] use univariate partitioning. To our knowledge, a comprehensive evaluation of bivariate partitioning does not exist yet. Karuppiah and Grossmann[2] mentioned the possibility of using bivariate partitioning, but preferred univariate partitioning for their study. They argued that the additional binary and continuous variables in bivariate partitioning might unacceptably increase the computational effort. Wicaksono and Karimi[27] reported improved relaxation quality from bivariate partitioning for a simple benchmark problem, however did not perform an extensive numerical comparison between univariate and bivariate partitioning. Although bivariate partitioning does increase the size of the MILP relaxation model, the size of the relaxation model is not the only factor that affects the performance of a GO algorithm. The quality of relaxation from a larger model may be better than that from a smaller model. The use of the larger model in a sBB-based GO algorithm may result in fewer nodes or iterations and less computation time to reach global optimality. As noted by Wicaksono and Karimi,[26] the computational effort for obtaining a piecewise linear relaxation varies with the MILP formulation for the relaxation. Thus, developing efficient and tighter formulations for univariate and bivariate partitioning and evaluating their performance numerically are of interest.

As pointed out by Misener et al.,[29] formulations based on the special ordered sets (SOS)[30] can be effective in obtaining piecewise linear relaxations for BLPs. They compared four formulations [linear segmentation, convex hull, classic convex combination (CC) with explicit binary variables, and SOS] for piecewise linear approximation of nonlinear functions for gas lifting operations. They found the formulation using the SOS2 (special ordered set of type 2)[31] variables to be the best computationally. However, their study targeted general nonlinear functions rather than just bilinear terms. Recently, Gounaris et al.[32] compared several univariate piecewise relaxation formulations for several pooling problems, some of which were similar to the ones developed by Wicaksono and Karimi.[26] They proposed an interesting idea of using SOS1 (special ordered set of type 1) variables to partition the variable domains and showed it to be computationally attractive for the piecewise relaxation of bilinear terms. However, they did not study bivariate partitioning and relied on the direct declaration of SOS variables in optimization solvers to implement SOS properties.

In this article, we present and evaluate various formulations using univariate and bivariate partitioning for the MILP relaxation of BLPs (and thus MIBLPs). First, we present some simple results for selecting the partitioned variables in univariate partitioning and the optimal choice of segment lengths. Next, we present 10 MILP relaxation models using the incremental cost (IC), CC, and SOS formulation approaches for both univariate and bivariate partitioning. Finally, we use four large process synthesis problems to evaluate them numerically.

Problem Statement

Consider the following BLP.

$$\begin{aligned}&\text{Minimize}\, f(x,z)\\ &\text{s.t.}\ \boldsymbol{g}(\boldsymbol{x},z)\le 0, \boldsymbol{h}(\boldsymbol{x},z)=0\\ &z_{ij}=x_ix_j \qquad\qquad (i,j)\in\mathbf{B}\\ &\boldsymbol{x}^{\mathrm{L}}\le\boldsymbol{x}\le\boldsymbol{x}^{\mathrm{U}}\end{aligned}$$

where $\boldsymbol{x}$ is a vector of I ($i = 1, \ldots, I$) continuous variables, z_{ij} represents the bilinear product of x_i and x_j, $\mathbf{B} = \{(i, j) \mid z_{ij} = x_ix_j\}$, $f(\boldsymbol{x}, z)$ is linear scalar function, and $\boldsymbol{h}(\boldsymbol{x}, z)$ and $\boldsymbol{g}(\boldsymbol{x}, z)$ are linear vector functions. With no loss of generality, we represent the above BLP as follows.

$$\begin{aligned}&\text{Minimize}\, f(x,z)\\ &\text{s.t.}\ \boldsymbol{g}(\boldsymbol{x},z)\le 0, \boldsymbol{h}(\boldsymbol{x},z)=0\\ &z_{ij}=x_ix_j \qquad\qquad (i,j)\in\mathbf{B}\\ &0\le\boldsymbol{x}\le 1\end{aligned}$$

As discussed earlier, we may have multiple bilinear terms in any given problem, and two options exist for relaxing each of them. One is to use LP relaxation, and the other is to use a piecewise relaxation. Which combination of the two relaxation strategies is the best for a given problem is still an open question that is beyond the scope of this work. In this article, we obtain a piecewise linear relaxation of $\mathbf{S} = \{(\boldsymbol{x}, \boldsymbol{z}) | z_{ij} = x_i x_j, (i,j) \in \mathbf{B}, 0 \le \boldsymbol{x} \le 1\}$ with the assumption that only piecewise relaxation is used for each bilinear term.

Partitioning

The first step in developing a piecewise linear relaxation is to partition one or more variables in a bilinear term. The univariate (bivariate) strategy partitions only one (both) of the two variables in a bilinear term. However, if two variables appear in a bilinear term, and both are partitioned, then it is not necessary to use bivariate partitioning for relaxing that bilinear term. One can still use univariate partitioning for that term. To illustrate this, consider a BLP with three bilinear terms: $z_{12} = x_1 x_2$, $z_{23} = x_2 x_3$, and $z_{31} = x_3 x_1$. We can achieve the univariate piecewise relaxation of all three terms by partitioning (x_1 and x_2), (x_1 and x_3), or (x_2 and x_3). It is clear that for each choice, one bilinear term will have both its variables partitioned. Thus, we have two choices for relaxing that term. We can select any of the two variables and relax that term using the univariate strategy, or we can use the bivariate strategy. Thus, the issue of identifying the best mix of univariate and bivariate strategies is unaddressed. While the literature so far has used no partitioning or univariate partitioning for all bilinear terms, bivariate partitioning or mixed partitioning (both univariate and bivariate) schemes have not been used. In this article, we do not address mixed strategy, but use either univariate or bivariate piecewise relaxation strategy uniformly across all bilinear terms.

We now address the question of how many and which variables one should partition. For any given problem, a feasible solution to the following IP gives us the minimum-cardinality set of partitioned variables, assuming that one uses a piecewise relaxation (univariate or bivariate) for each bilinear term.

$$\text{Minimize } \sum_{i=1}^{I} y_i \text{ subject to } y_i + y_j \ge 1 \text{ for each } (i,j) \in \mathbf{B}$$

$$\text{where } y_i = \begin{cases} 1 & \text{if } i \text{ is partitioned} \\ 0 & \text{otherwise} \end{cases}$$

If $\Pi = \{i \mid x_i \textit{ is not partitioned}\}$, then Π is nonempty (empty) for univariate (bivariate) partitioning. After selecting the variables to partition, we must decide how to partition. Let us partition x_i, $i \notin \Pi$, $0 \le x_i \le 1$, into N_i arbitrary, exclusive, and exhaustive segments using N_i+1 grid points (a_{in}, $n = 0, 1, \ldots, N_i$, $a_{i0} = 0$, $a_{iN_i} = 1$). Denote $d_{in} = a_{in} - a_{i(n-1)}$ as the length of segment n $\{[a_{i(n-1)}, a_{in}], n = 1, \ldots, N_i\}$. The simplest option for positioning these grid points is to place them uniformly in [0,1], i.e., to use identical segment lengths. We may term this as uniform placement as opposed to nonuniform placement (nonidentical segment lengths). Although uniform placement seems to be the simplest, the criteria for and identification of optimal placement have not been addressed. We now show that uniform placement is in fact optimal with respect to one simple criterion. However, this may not necessarily be optimal from the perspective of solving the BLP efficiently.

The LP relaxation[19] of $\{z_{ij} = x_i x_j, 0 \le z_{ij}, x_i, x_j \le 1\}$ has the following convex (Eq. 1) and concave (Eqs. 2) linear estimators.

$$z_{ij} \ge x_i + x_j - 1 \qquad (i,j) \in \mathbf{B} \tag{1}$$

$$z_{ij} \le x_i \qquad (i,j) \in \mathbf{B} \tag{2a}$$

$$z_{ij} \le x_j \qquad (i,j) \in \mathbf{B} \tag{2b}$$

Androulakis et al.[33] showed that the maximum separation of z_{ij} from its convex underestimators ($z_{ij} \ge 0$ and Eq. 1) is 1/4 and occurs at $x_i = x_j = 1/2$. Appendix A generalizes the same result for the the entire convex and concave envelope ($z_{ij} \ge 0$, Eqs. 1 and 2).

Consider an arbitrary segment n of x_i ($i \notin \Pi$) for the univariate partitioning of $\{z_{ij} = x_i x_j, 0 \le x_i, x_j \le 1\}$. The maximum separation of z_{ij} from the LP relaxation for this segment is $d_{in}/4$. For the "best" partitioning, let us minimize the sum of squares of these separations for all the N_i segments. This gives us the following optimization problem.

$$\text{Minimize } \sum_{n=1}^{N_i} \frac{d_{in}^2}{16} \text{ subject to } \sum_{n=1}^{N_i} d_{in} = 1$$

The optimal solution for the above (Appendix B) is the uniform placement ($d_{in} = 1/N_i$), which holds for bivariate partitioning (Appendix B) as well.

Lemma I

The uniform placement of grid points for both univariate and bivariate partitioning is a scheme that minimizes the sum of squares of the maximum separation of $z_{ij} = x_i x_j$ from its LP relaxation in each segment.

In the absence of any other easy justification for selecting the best placement strategy, we use $d_{in} = d_i = 1/N_i$ in this work based on the aforementioned result. Thus, $a_{in} = nd_i = n/N_i$. Every value of x_i must fall in one of the N_i partitions. The literature[1,2,26,32] has used several approaches for modeling this basic fact and developed various formulations for piecewise linear relaxation. Wicaksono and Karimi[26] compared three alternate formulations based on univariate partitioning, namely big-M, CC, and IC.[34] They concluded that the big-M approach can exhibit poor relaxation quality and is not competitive. Therefore, we do not use the big-M approach in this work. In contrast, IC and CC models represent convex hulls, but differ in solution speed. In this work, we develop and compare several new univariate and bivariate formulations for the MILP relaxation of $\mathbf{S} = \{(\boldsymbol{x}, \boldsymbol{z}) \mid z_{ij} = x_i x_j, (i, j) \in \mathbf{B}, 0 \le \boldsymbol{x} \le 1\}$ using the IC, CC, and SOS approaches. For completeness, we also include the best IC and CC univariate formulations of Wicaksono and Karimi.[26]

Incremental Cost Formulations

In this approach, the following binary variable[26,30,34] is used to model x_i.

$$\mu_{in} = \begin{cases} 1 & \text{if } x_i \ge nd_i \\ 0 & \text{otherwise} \end{cases} \qquad i \notin \Pi, 1 \le n \le N_i - 1$$
$$\mu_{in} \ge \mu_{i(n+1)} \qquad i \notin \Pi, 1 \le n \le N_i - 2 \tag{3}$$

Wicaksono and Karimi[26] introduced the use of global differential variables to model x_i in conjunction with μ_{in} and presented a formulation (**NF12**) for univariate partitioning with uniform placement. **NF12** in terms of our notation is as follows:

$$x_i = d_i \sum_{n=1}^{N_i-1} \mu_{in} + \Delta x_i \qquad i \notin \Pi \tag{4a}$$

$$0 \le \Delta x_i \le d_i \qquad i \notin \Pi \tag{4b}$$

$$z_{ij} = d_i \sum_{n=1}^{N_i-1} \Delta v_{ijn} + \Delta z_{ij} \qquad (i,j) \in \mathbf{B}, i \notin \Pi, j \in \Pi \tag{5}$$

$$0 \le \Delta v_{ij(N_i-1)} \le \Delta v_{ij(N_i-2)} \le \ldots \le \Delta v_{ij2} \le \Delta v_{ij1} \le x_j \qquad (i,j) \in \mathbf{B}, i \notin \Pi, j \in \Pi \tag{6}$$

$$\Delta v_{ij1} \ge \mu_{i1} + x_j - 1 \qquad (i,j) \in \mathbf{B}, i \notin \Pi, j \in \Pi \tag{7a}$$

$$\Delta v_{ijn} \ge \left(\mu_{in} - \mu_{i(n-1)}\right) + \Delta v_{ij(n-1)} \qquad (i,j) \in \mathbf{B}, i \notin \Pi, j \in \Pi, 2 \le n \le N_i - 1 \tag{7b}$$

$$\Delta v_{ij(N_i-1)} \le \mu_{i(N_i-1)} \qquad (i,j) \in \mathbf{B}, i \notin \Pi, j \in \Pi \tag{7c}$$

$$\Delta z_{ij} \le \Delta x_i \qquad (i,j) \in \mathbf{B}, i \notin \Pi, j \in \Pi \tag{8a}$$

$$\Delta z_{ij} \le d_i x_j \qquad (i,j) \in \mathbf{B}, i \notin \Pi, j \in \Pi \tag{8b}$$

$$\Delta z_{ij} \ge \Delta x_i + d_i\left(x_j - 1\right) \qquad (i,j) \in \mathbf{B}, i \notin \Pi, j \in \Pi \tag{8c}$$

We call the above model **U-IC**, where **U** signifies univariate partitioning.

Following the approach of Wicaksono and Karimi,[26] we now develop a formulation analogous to **U-IC** for bivariate partitioning. Using Eqs. 3 and 4, we express,

$$z_{ij} = d_i d_j \sum_{n=1}^{N_i-1} \sum_{m=1}^{N_j-1} \theta_{ijmn} + \sum_{n=1}^{N_i-1} d_i \Delta v_{ijn} + \sum_{m=1}^{N_j-1} d_j \Delta v_{jim} + \Delta z_{ij} \qquad (i,j) \in \mathbf{B} \tag{9}$$

where $\theta_{ijmn} = \mu_{in}\mu_{jm}$, $\Delta v_{ijn} = \mu_{in}\Delta x_j$, and $\Delta z_{ij} = \Delta x_i \Delta x_j$. We linearize $\theta_{ijmn} = \mu_{in}\mu_{jm}$ by using the following with $\theta_{ijmn} \ge 0$,

$$\theta_{ijmn} \ge \mu_{in} + \mu_{jm} - 1 \qquad (i,j) \in \mathbf{B}, 1 \le n \le N_i - 1, 1 \le m \le N_j - 1 \tag{10a}$$

$$\theta_{ijmn} \le \mu_{in} \qquad (i,j) \in \mathbf{B}, 1 \le n \le N_i - 1, 1 \le m \le N_j - 1 \tag{10b}$$

$$\theta_{ijmn} \le \mu_{jm} \qquad (i,j) \in \mathbf{B}, 1 \le n \le N_i - 1, 1 \le m \le N_j - 1 \tag{10c}$$

For linearizing $\Delta v_{ijn} = \mu_{in}\Delta x_j$, we use the bounds [$\mu_{i2}$, 1] for μ_{i1}, [$\mu_{i(n+1)}$, $\mu_{i(n-1)}$] for μ_{in} (n = 2 to N_i−1), [0, $\mu_{i(N-1)}$] for μ_{iN}, and [0, d_j] for Δx_j to obtain,

$$0 \le \Delta v_{ij(N_i-1)} \le \Delta v_{ij(N_i-2)} \le \ldots \le \Delta v_{ij2} \le \Delta v_{ij1} \le \Delta x_j \quad (i,j) \in \mathbf{B} \tag{11}$$

$$\Delta v_{ij1} \ge d_j \mu_{i1} + \Delta x_j - d_j \qquad (i,j) \in \mathbf{B} \tag{12a}$$

$$\Delta v_{ijn} \ge d_j\left[\mu_{in} - \mu_{i(n-1)}\right] + \Delta v_{ij(n-1)} \qquad (i,j) \in \mathbf{B}, 2 \le n \le (N_i - 1) \tag{12b}$$

$$\Delta v_{ij(N_i-1)} \le d_j \mu_{i(N_i-1)} \qquad (i,j) \in \mathbf{B} \tag{12c}$$

Lastly, to linearize $\Delta z_{ij} = \Delta x_i \Delta x_j$ for $(i, j) \in \mathbf{B}$, we use the following:

$$\Delta z_{ij} \le d_i \Delta x_j \qquad (i,j) \in \mathbf{B} \tag{13a}$$

$$\Delta z_{ij} \le d_j \Delta x_i \qquad (i,j) \in \mathbf{B} \tag{13b}$$

$$\Delta z_{ij} \ge d_i \Delta x_j + d_j \Delta x_i - d_i d_j \qquad (i,j) \in \mathbf{B} \tag{13c}$$

This completes our model **B-IC** (Eqs. 3, 4, 9–13) for bivariate partitioning, where **B** signifies bivariate partitioning.

Convex Combination Formulations

The best CC formulation (**NF11**) from Wicaksono and Karimi[26] based on global incremental variables uses the following binary variable and constraints along with Eqs. 4b and 8.

$$\lambda_{in} = \begin{cases} 1 & \text{if } (n-1)d_i \le x_i < n d_i \\ 0 & \text{otherwise} \end{cases} \qquad i \notin \Pi, 1 \le n \le N_i$$

$$\sum_{n=1}^{N_i} \lambda_{in} = 1 \qquad i \notin \Pi \tag{14}$$

$$x_i = d_i \sum_{n=1}^{N_i} (n-1)\lambda_{in} + \Delta x_i \qquad i \notin \Pi \tag{15a}$$

$$x_j = \sum_{n=1}^{N_j} \Delta y_{jn} \qquad j \in \Pi \tag{15b}$$

$$z_{ij} = d_i \sum_{n=1}^{N_i} (n-1)\Delta y_{jn} + \Delta z_{ij} \qquad (i,j) \in \mathbf{B}, i \notin \Pi, j \in \Pi \tag{16}$$

$$0 \le \Delta y_{jn} \le \lambda_{in} \qquad (i,j) \in \mathbf{B}, i \notin \Pi, j \in \Pi \tag{17}$$

We call the above model **U-CC** (Eqs. 4b, 8, 14–17).

As we did for IC, we obtain a model **B-CC** for bivariate partitioning analogous to **U-CC** as follows. Using Eqs. 14 and 15, we obtain,

$$z_{ij} = d_i d_j \sum_{n=1}^{N_i} \sum_{m=1}^{N_j} (m-1)(n-1)\delta_{ijmn} + \sum_{m=1}^{N_i} (n-1) d_i \Delta y_{ijn} + \sum_{m=1}^{N_j} (m-1) d_j \Delta y_{jim} + \Delta z_{ij} \qquad (i,j) \in \mathbf{B} \tag{18}$$

where $\delta_{ijmn} = \lambda_{in}\lambda_{jm}$, $\Delta z_{ij} = \Delta x_i \Delta x_j$, and $\Delta y_{ijn} = \lambda_{in}\Delta x_j$ for $(i, j) \in \mathbf{B}$. We linearize[35] $\delta_{ijmn} = \lambda_{in}\lambda_{jm}$ and $\Delta y_{ijn} = \lambda_{in}\Delta x_j$ by using the following with $\delta_{ijmn} \geq 0$ and $\Delta y_{ijn} \geq 0$.

$$\sum_{m=1}^{N_j} \delta_{ijmn} = \lambda_{in} \qquad (i,j) \in \mathbf{B}, 1 \leq n \leq N_i \tag{19a}$$

$$\sum_{n=1}^{N_i} \delta_{ijmn} = \lambda_{jm} \qquad (i,j) \in \mathbf{B}, 1 \leq m \leq N_j \tag{19b}$$

$$\sum_{n=1}^{N_i} \Delta y_{ijn} = \Delta x_j \qquad (i,j) \in \mathbf{B} \tag{20a}$$

$$\Delta y_{ijn} \leq d_j \lambda_{in} \qquad (i,j) \in \mathbf{B}, 1 \leq n \leq N_i \tag{20b}$$

$$\Delta y_{ijn} \geq d_j(\lambda_{in} - 1) + \Delta x_j \qquad (i,j) \in \mathbf{B}, 1 \leq n \leq N_i \tag{20c}$$

Then, including Eqs. 13–15 to linearize $\Delta z_{ij} = \Delta x_i \Delta x_j$, we get model **B-CC** (Eqs. 13–15 and 18–20) for bivariate partitioning.

In addition to the IC and CC approaches, which both make use of explicit binary variables, a third approach is to express x_i as a CC of grid points using SOS variables. We now present models based on this approach.

SOS Formulations

This approach expresses x_i as follows:

$$x_i = d_i \sum_{n=1}^{N_i} n\zeta_{in} \qquad i \notin \Pi \tag{21a}$$

$$\sum_{n=0}^{N_i} \zeta_{in} = 1 \qquad i \notin \Pi \tag{21b}$$

where $0 \leq \zeta_{in} \leq 1$ are SOS2 variables, i.e., at most two of them can be positive, and the two must be adjacent.

For univariate partitioning, we obtain the following using Eqs. 21.

$$z_{ij} = d_i \sum_{n=1}^{N_i} n w_{ijn} \qquad (i,j) \in \mathbf{B}, i \notin \Pi, j \in \Pi \tag{22}$$

where $w_{ijn} = \zeta_{in} x_j$ for $(i, j) \in \mathbf{B}$, $i \notin \Pi$, $j \in \Pi$. Then, we use the following constraints to linearize this.

$$w_{ijn} \leq \zeta_{in} \qquad (i,j) \in \mathbf{B}, i \notin \Pi, j \in \Pi \tag{23a}$$

$$\sum_{n=0}^{N_i} w_{ijn} = x_j \qquad (i,j) \in \mathbf{B}, i \notin \Pi, j \in \Pi \tag{23b}$$

$$w_{ijn} \geq \zeta_{in} + x_j - 1 \qquad (i,j) \in \mathbf{B}, i \notin \Pi, j \in \Pi \tag{23c}$$

Equations 21–23 and $w_{ijn} \geq 0$ constitute the formulation using SOS2 variables for univariate partitioning. We call this model **U-SOS2-I**, where **I** signifies that SOS2 variables are handled implicitly by the solver. GAMS/CPLEX[36] accepts and solves models with SOS2 variables by using binary variables internally. Balas[37] proved that SOS2 formulations represent the convex hull.

Although GAMS/CPLEX uses binary variables internally to handle SOS2 variables, our experience suggests that handling SOS2 constraints using explicit binary variables may be better in some instances. Therefore, we use the following approach proposed by Keha et al.[30]:

$$\eta_{in} = \begin{cases} 1 & \text{if only } \zeta_{in} \text{ and } \zeta_{i(n+1)} \text{ are positive} \\ 0 & \text{otherwise} \end{cases} \qquad i \notin \Pi, 0 \leq n \leq N_i - 1$$

$$\sum_{n=0}^{N_i-1} \eta_{in} = 1 \qquad i \notin \Pi \tag{24}$$

$$\zeta_{i0} \leq \eta_{i0} \qquad i \notin \Pi \tag{25a}$$

$$\zeta_{in} \leq \eta_{i(n-1)} + \eta_{in} \qquad i \notin \Pi, 1 \leq n \leq N_i - 1 \tag{25b}$$

$$\zeta_{iN_i} \leq \eta_{i(N_i-1)} \qquad i \notin \Pi \tag{25c}$$

Note that ζ_{in} is now just an ordinary continuous variable. We call this model (Eqs. 21–25) **U-SOS2-E**, where **E** signifies the use of explicit binary variables to handle SOS2 variables.

While **U-SOS2-E** treats η_{in} as binary, we can also declare them as SOS1 variables, and let GAMS/CPLEX handle them implicitly. Thus, we have an alternate model that is the same as **U-SOS2-E**, but binary variables are treated as SOS1 variables. We call this model **U-SOS1-I**.

In the above, we presented three univariate models (**U-SOS2-I**, **U-SOS2-E**, and **U-SOS1-I**) based on the SOS approach. As done earlier for other models, we can derive analogous bivariate SOS models, namely **B-SOS2-I**, **B-SOS2-E**, and **B-SOS1-I**. All three models use Eqs. 26 and 27 instead of Eqs. 22 and 23.

$$z_{ij} = d_i d_j \sum_{n=1}^{N_i} \sum_{m=1}^{N_j} mn\omega_{ijmn} \qquad (i,j) \in \mathbf{B} \tag{26}$$

$$\sum_{m=0}^{N_j} \omega_{ijmn} = \zeta_{in} \qquad (i,j) \in \mathbf{B}, 0 \leq n \leq N_i \tag{27a}$$

$$\sum_{n=0}^{N_i} \omega_{ijmn} = \zeta_{jm} \qquad (i,j) \in \mathbf{B}, 0 \leq m \leq N_j \tag{27b}$$

The 10 models (**U-IC**, **B-IC**, **U-CC**, **B-CC**, **U-SOS2-I**, **U-SOS2-E**, **U-SOS1-I**, **B-SOS2-I**, **B-SOS2-E**, and **B-SOS1-I**) presented above have different model sizes. All represent the convex hull,[26,37] but may differ in computational speed. The bivariate models would be larger than their univariate counterparts, but may yield higher piecewise gain (PG).[26] We now evaluate these models numerically for several case studies. The set of partitioned variables for each case study was determined by solving the IP presented earlier.

Case Studies

We present four case studies. The first (MIBLP) is the synthesis of heat exchanger networks (HENS). The second

Table 1. Stream Data for Case Study 1

Stream	Initial Temperature (°C)	Final Temperature (°C)	Heat Capacity Flowrate (kW/C)
H1	180	75	30
H2	240	60	40
C1	40	230	35
C2	120	300	20
Cold utility	25	40	–
Hot utility	325	325	–

(MIBLP) is the generalized pooling problem from Meyer and Floudas.[1] The third (BLP) is the synthesis of integrated water-using and water-treating networks from Karuppiah and Grossmann.[2] The fourth (BLP) is a nonsharp distillation column sequencing problem from Floudas et al.[11]

For all runs, we used a Dell Precision AW-T7400 with Quad-Core Intel® Xeon® X5492 (3.4 GHz) Processor, 64 GB of RAM, Windows XP Professional x64, GAMS 22.8, CPLEX v.11.1.1 as the LP and MILP solver, CONOPT v.3 and MINOS v.5.51 as the NLP solvers, and BARON v.7.5 and DICOPT as the MIBLP solvers. We used $N_i = 2$, 3, and 4 for all case studies and set the relative gap tolerance to zero for all runs to achieve optimality. We set 5000 CPU s as the upper limit for each run. If a model fails to reach an optimal solution within this time, then we take 5000 CPU s as its solution time.

GAMS/CPLEX allows one to specify branching priorities for the binary variables. In solving MIBLPs, we have two types of binary variables. One belongs to the original MIBLP model, and the other is used to model partitioning and piecewise linear relaxation. The second type includes the SOS and binary variables (μ_{in}, λ_{in}, ζ_{in}). Only one of these variables appears in each model. We observed that giving priorities to these variables for branching reduces the solution times drastically. Therefore, we use the prioropt option in GAMS to specify that SOS or binary variables (μ_{in}, λ_{in}, ζ_{in}) must be branched first while solving a model. We do not assign any priority to the binary variables that are not meant for partitioning.

Case study 1: HENS

If one allows stream splitting in a HENS problem and relaxes the assumption of isothermal mixing to include more alternatives, then the optimization formulation involves bilinear terms involving the products of flow and temperature and heat transfer area and temperature. Appendix C presents such a MIBLP model for HENS, which we use as the base formulation for this case study. The model uses a two-stage superstructure[38] of two hot (H1 and H2) and two cold (C1 and C2) streams. In each stage, splits of process streams (hot or cold) exchange heat using 2-stream exchangers. Utility-based coolers and heaters are at the ends of the superstructure. Table 1 gives the stream and cost data.

The MIBLP model has linear objective and constraints, except the energy balances and heat transfer equations that involve bilinear terms. It has 12 binary variables, 88 continuous variables, 28 nonlinear constraints, and 52 bilinear terms. DICOPT with default initialization scheme in GAMS fails to give a feasible solution for this problem. BARON with the default starting point keeps on iterating for more than 5000 CPU s with an initial lower bound (LB) of 1155.25 and upper bound (UB) of 3,288,000. The bilinear terms involve 72 variables and we select 28 variables (A_{hck}, Acu_h, Ahu_c, fh_{hck}, and fc_{hck}) for univariate partitioning.

Case study 2

This is the generalized pooling problem on wastewater treatment networks from Meyer and Floudas.[1] We refer to the work of Meyer and Floudas as MF. The case study involves 7 source nodes and 1 sink node for effluents. The goal is to reduce three contaminants in the source streams before the effluent can be discharged to the sink. The superstructure (Figure 1 of MF) has 10 wastewater treatment plants with various technologies. Appendix D presents the MIBLP model that forms the basis for this case study. Tables MF-A1 to MF-A9 have the relevant data. Because MF does not provide all the variable bounds, we set $\overline{a}_s = \overline{d}_{st} = f_s^{\mathrm{source}}$ and $\overline{b}_t$, $\overline{c}_{tt'}$, and $\overline{e}_t$ to be the sum of all f_s^{source} based on our understanding of the problem. In this case study, LB is very sensitive to the UB of q_{ct}, which is the quality of contaminant c in effluent t. Conservatively, we set this UB as the sum of the qualities at the source.

We use the formulation of MF for a single sink. It consists of two types of bilinear terms. One involves the products of the quality (q_{ct}) and the input flows to plants from sources (d_{st}) and other plants ($c_{t't}$). These bilinear terms are present in Eqs. 16 and 17 of MF. Following Tawarmalani et al.[39] and Liberti and Pantelides,[40] we consider any product involving one continuous variable and a sum of several continuous variables as sums of bilinear terms of two variables. In other words, we replace $q_{ct}\sum_{s\in S} d_{st}$ by $\sum_{s\in S} q_{ct}d_{st}$ and $q_{ct}\sum_{t'\in T\setminus(t)} c_{t't}$ by $\sum_{t'\in T\setminus(t)} q_{ct}c_{t't}$. This improves the relaxation quality.[39] The selection of treatment plants and the existence of various network streams are modeled using binary variables, which result in a large MIBLP. This MIBLP has 187 binary variables, 190 continuous variables, 33 nonlinear constraints, and 1290 bilinear terms. For univariate models, we partition all the 30 quality variables. As in the previous case study, DICOPT cannot solve even the relaxed MINLP, and BARON with the default starting point keeps on iterating with an initial LB of 102,766 and UB of 1,386,980.

Case study 3

The third case study (Example 4 of Karuppiah and Grossmann[2]) involves the synthesis of integrated water use and treatment systems. We use KG to refer to that work and adopt their notation for equations, figures, tables, and sections. Appendix E gives the BLP model from KG, which we use as the base formulation for this case study. 5 process units (PU1-5) in the network consume fresh or treated water and generate water with three contaminants (A, B, C). This contaminated water is treated in three treatment units (TU1-3). Tables KG-7 and KG-8 in section KG-7.4 list the numerical data for this case study. The superstructure (Figure KG-17) has 9 splitters (SU1-9) and 9 mixers (MU1-9). Because MU1-5 supply water at fixed flows to PU1-5, the UBs on the flows to MU1-5 are set at these fixed flows. Similarly,

Table 2. Model Statistics for the Case Studies

Partition Type		Univariate					Bivariate				
		IC	CC	SOS			IC	CC	SOS		
Model Type	N_i	U-IC	U-CC	U-SOS2-I	U-SOS2-E	U-SOS1-I	B-IC	B-CC	B-SOS2-I	B-SOS2-E	B-SOS1-I
Case study 1											
Binary variables	2	40	68	12	68	12	86	160	12	160	12
	3	68	96	12	96	12	160	234	12	234	12
	4	96	124	12	124	12	234	308	12	308	12
Continuous variables	2	276	326	430	430	486	426	686	834	834	982
	3	328	362	510	510	594	686	1050	1272	1272	1494
	4	380	408	590	590	702	1050	1518	1814	1814	2110
Constraints	2	497	519	623	735	735	855	1189	617	913	913
	3	629	571	727	867	867	1605	1501	721	1091	1091
	4	761	623	831	999	999	2667	1813	825	1269	1269
Nonzeros	2	1341	1379	1777	2029	2029	2311	3343	2423	3089	3089
	3	1789	1637	2197	2561	2561	4509	4999	3715	4677	4677
	4	2237	1895	2617	3093	3093	7539	6967	5319	6577	6577
Case study 2											
Binary variables	2	217	247	187	247	187	387	587	187	587	187
	3	247	277	187	277	187	587	787	187	787	187
	4	277	307	187	307	187	787	987	187	987	187
Continuous variables	2	2028	1860	2770	2770	2830	3220	5770	6170	6170	6570
	3	2538	2030	3310	3310	3400	5770	9340	9940	9940	10540
	4	3048	2200	3850	3850	3970	9340	13930	14730	14730	15530
Constraints	2	4096	3788	4808	4928	4928	7328	10078	4468	5268	5268
	3	5144	4298	5628	5978	5978	14158	13138	5488	6488	6488
	4	6192	4808	6848	7028	7028	24048	16198	6508	7708	7708
Nonzeros	2	12811	11857	15797	16067	16067	21487	30557	20047	21847	21847
	3	16465	13617	19427	19817	19817	41467	45747	31157	33757	33757
	4	20119	15377	23057	23567	23567	69607	63997	45327	48727	48727
Case study 3											
Binary variables	2	73	146	0	146	0	344	688	0	688	0
	3	146	219	0	219	0	688	1032	0	1032	0
	4	219	292	0	292	0	1032	1376	0	1376	0
Continuous variables	2	1089	1308	1658	1658	1804	1978	2893	3566	3956	4644
	3	1308	1527	1950	1958	2170	3268	5074	6106	6106	7138
	4	1527	1746	2242	2242	2534	5074	6498	8772	8774	10148
Constraints	2	1873	1946	2384	2676	2676	3965	4897	2488	4137	4137
	3	2384	2165	2822	3187	3187	7663	7147	3277	4997	4997
	4	2895	2384	3260	3698	3698	12909	7525	3793	5857	5857
Nonzeros	2	4779	4925	6531	7188	7188	10362	13504	9246	13461	13461
	3	6531	5947	8210	9159	9159	21177	23579	16453	20925	20925
	4	8283	6969	9889	11130	11130	36114	28899	24083	29931	29931
Case study 4											
Binary variables	2	6	12	0	12	0	72	144	0	144	0
	3	12	18	0	18	0	144	216	0	216	0
	4	18	24	0	24	0	216	288	0	288	0
Continuous variables	2	65	61	93	93	105	205	265	409	409	553
	3	77	65	111	111	129	265	349	565	565	781
	4	89	69	129	129	153	349	457	745	745	1031
Constraints	2	107	105	129	153	153	245	377	245	533	533
	3	137	117	153	183	183	473	449	269	629	629
	4	167	129	177	213	213	773	521	293	725	725
Nonzeros	2	274	262	360	414	414	664	1012	833	1481	1481
	3	376	314	456	534	534	1336	1504	1229	2165	2165
	4	478	366	552	654	654	2200	2068	1697	2921	2921

the UBs on the split flows from SU2-6 are also set at the same fixed flows. The maximum discharge limit for all contaminants is 10 ppm. Because PU1-5 have upper limits on the allowable contaminant flows, the UBs on the contaminant flows are set to these limits. For the remaining water and contaminant flows, we use the total flow to PU1-5 and 100 ppm, respectively, as the UBs. All LBs are set to zero.

We take the model of KG as the base formulation, but with a linear objective, which is to minimize the fresh water consumption and total flow to TU1-3. We use Eqs. KG-1a, 2–9, and 15 and simplify the KG model further by reducing several variables and constraints. First, we replace the total fresh water flow variables by the total split flows from SU1 in Eq. KG-4. Second, we assume the fresh water to be contaminant-free, hence eliminate the part of Eq. KG-5 for SU1. Similarly, we assume the fresh water to MU1-5 also to be contaminant-free. We treat Eq. KG-6 as a bound. We eliminate 15 bilinear terms by replacing the flows to PU1-5 by their fixed values in Eq. KG-3. Note that Eq. KG-3 modeling the individual contaminant balances is the only source of

Table 3. Solution Statistics for the Case Studies

		Univariate					Bivariate				
Partition Type		IC	CC	SOS			IC	CC	SOS		
Model Type	N_i	U-IC	U-CC	U-SOS2-I	U-SOS2-E	U-SOS1-I	B-IC	B-CC	B-SOS2-I	B-SOS2-E	B-SOS1-I
Case study 1											
CPU time (s)	2	0.203	0.562	0.187	0.203	0.203	0.874	0.968	1.203	0.687	0.421
	3	0.218	0.203	0.265	0.312	0.313	4.406	4.578	46.265	3.156	1.843
	4	0.218	0.765	0.296	0.531	0.431	17.921	10.281	2676.837	5.125	14.559
Nodes	2	40	1191	98	80	80	703	871	2277	593	424
	3	120	122	194	214	214	1791	1609	145629	2592	1446
	4	96	709	305	490	490	4366	3116	2901048	3697	19065
Case study 2											
CPU time (s)	2	0.921	0.687	0.937	0.593	0.59	2.296	5.359	2.75	1.421	1.406
	3	0.765	0.984	1.187	0.937	0.918	10.062	7.265	4.703	2.734	2.765
	4	0.984	1.124	1.484	0.984	0.981	21.547	14.421	48.922	13.65	12.718
Nodes	2	80	56	105	58	58	80	134	332	77	77
	3	57	93	103	77	77	154	100	317	131	131
	4	50	118	92	66	66	113	107	1054	230	230
Case study 3											
CPU time (s)	2	0.14	0.171	0.421	0.14	0.156	471.628	5.64	5000	45.469	77.753
	3	0.187	0.203	1.109	0.156	0.171	5000	5000	5000	5000	1390
	4	0.171	0.203	2.203	1.609	0.64	5000	5000	5000	5000	5000
Nodes	2	1	1	141	1	1	14371	700	–	995	997
	3	1	1	186	1	1	–	–	–	–	35032
	4	1	1	205	30	1	–	–	–	–	–
Case study 4											
CPU time (s)	2	0.015	0.093	0.046	0.015	0.001	0.109	0.124	0.015	0.109	0.125
	3	0.093	0.078	0.093	0.015	0.015	0.203	0.187	0.015	0.124	0.14
	4	0.093	0.093	0.109	0.093	0.015	0.203	0.203	0.062	0.187	0.14
Nodes	2	1	1	5	1	1	6	7	53	8	8
	3	1	1	5	1	1	10	8	42	25	25
	4	1	1	5	1	1	17	1	58	31	31

bilinearity (stream flow $\times$ contaminant concentration). The base model has 264 constraints, 344 variables (86 flow and 258 concentration), and 219 bilinear terms. While Example 4 in KG is a nonconvex NLP, our modification is a BLP. We partition the 86 flow variables in our univariate models. CONOPT and MINOS show infeasibility and fail to reach even a local solution to this BLP. BARON does not show infeasibility, but begins with a poor LB of 40 and UB of 262.2, which do not improve even after a long time.

Case study 4

The final case study involves a benchmark process network synthesis problem from Floudas et al.[11] It is a column-sequencing problem for nonsharp distillation, which was formulated as a BLP with 24 variables, 17 constraints, and 12 bilinear terms. The bilinear terms involve six flow and four composition variables. The UBs on all flow variables are set to 180 kg/(mol h) and the LB for the flow of stream-18 is set to 10 kg/(mol h) without cutting off the reported global optimal solution. For our univariate models, we partition the flow variables.

Results and Discussion

Table 2 lists the model statistics for the case studies. Table 3 gives the CPU times and branch-and-bound nodes for various runs. Overall, the univariate models are more efficient, require fewer nodes, often outperform other models by a clear distance, and work well for both BLPs and MIBLPs. However, they provide poorer relaxations when compared with bivariate models. SOS1 models seem to be the most competitive overall. **U-SOS1-I**, **U-IC**, and **B-SOS1-I** are the most efficient. Because the SOS variables seem to work better when they are prioritized for branching before the intrinsic binary variables of the MIBLPs in GAMS/CPLEX, the results in Table 3 involve the use of such a branching priority. Note that we assign priority (over the intrinsic binary variables) to the SOS variables only, and no priorities to the

Table 4. MILP Objective and Piecewise Gains (PG) for Univariate and Bivariate Partitioning

		Case Study 1		Case Study 2		Case Study 3		Case Study 4	
	N_i	Univariate	Bivariate	Univariate	Bivariate	Univariate	Bivariate	Univariate	Bivariate
MILP Objective	2	95018.6	116383.4	400956.5	406187.2	184.2	184.2	1.279	1.431
	3	100463.4	123899.9	410434.5	412197.5	184.2	190.5	1.279	1.431
	4	108613.4	134060.6	413210.2	414728.0	184.2	218.6	1.279	1.431
PG	2	0.001	0.226	0	0.013	0	0	0	0.119
	3	0.058	0.305	0.024	0.028	0	0.034	0	0.119
	4	0.144	0.412	0.031	0.034	0	0.187	0	0.119

Table 5. Relative CPU Times for Various Models with $N_i = 2$

Partition Type	Univariate					Bivariate				
Model Type	IC	CC	SOS			IC	CC	SOS		
Case Study\ Model	U-IC	U-CC	U-SOS2-I	U-SOS2-E	U-SOS1-I	B-IC	B-CC	B-SOS2-I	B-SOS2-E	B-SOS1-I
1	1.09	3.01	1	1.09	1.09	4.67	5.18	6.43	3.67	2.25
2	1.56	1.16	1.59	1.01	1	3.89	9.08	4.66	2.41	2.38
3	1	1.22	3.01	1	1.11	3368.8	40.3	35714.3	324.8	555.4
4	15	93	46	15	1	109	124	15	109	125

intrinsic variables. For BLPs (case studies 3 and 4), no prioritization is required. However, further detailed studies are required to optimize the branching strategies in GAMS for SOS formulations.

One or more SOS models outperform IC and CC models in each case study. Thus, the SOS formulations in general seem more attractive computationally than the IC and CC formulations.

Although **U-SOS2-I** and **B-SOS2-I** do not use explicit binary variables, they are not as efficient computationally as **U-SOS2-E** and **B-SOS2-E** in many instances. Thus, it is not always beneficial to use the implicit SOS2 structure for piecewise relaxation.

One major goal of piecewise relaxation is to improve the quality of relaxation over that of the LP relaxation. Therefore, it is crucial to measure the improvement or gain in the quality of relaxation. Wicaksono and Karimi[26] defined piecewise gain (PG) for this purpose as follows:

$$\mathrm{PG} = \frac{\text{MILP Objective} - \text{LP Objective}}{\text{LP Objective}} \tag{28}$$

PG = 0 means no gain from the piecewise relaxation over LP relaxation, with higher values being more desirable. The objective values from the LP relaxation are 94959.6, 400956.5, 184.2, and 1.279, respectively, for case studies 1–4. Table 4 lists the MILP objectives and PG values for each case study. As expected, they are the same for all models for a given partitioning scheme, but increase with N_i. Most importantly, for a given N_i, bivariate partitioning improves the MILP objective and gives a higher PG. For case study 1, PG for bivariate partitioning is as high as 0.412 for $N_i = 4$. Except for case study 2, the highest PG ($N_i = 4$) for univariate partitioning is even lower than that for bivariate partitioning with $N_i = 2$. Significantly, while no univariate model improves PG even with increasing N_i for case studies 3 and 4, bivariate models increase it each time. Note that case studies 3 and 4 are BLPs, and not MIBLPs. Overall, bivariate partitioning improves PG in all case studies, while univariate partitioning fails to do so for the two BLPs.

Considering the tradeoff between relaxation quality and computational performance, the case of $N_i = 2$ seems particularly interesting, as the bivariate models seem competitive with univariate models in terms of CPU times for $N_i = 2$. Their performance is consistent except for case study 3, where they fail to converge even after 5000 CPU s for $N_i > 2$. Table 5 gives the relative CPU times[41] with $N_i = 2$ for the 10 models. The relative CPU time defined for this purpose is as follows:

$$\text{Relative CPU time} = \frac{\text{CPU time for the current model}}{\text{Least CPU time from among the 10 models}} \tag{29}$$

These are computed based on the minimum CPU time by any model for given case study and N_i. Because the CPU times invariably increase with N_i, a 2-segment bivariate partitioning scheme offers an attractive compromise between relaxation quality and computation time.

We also compared **U-SOS1-I** and **B-SOS1-I** on case study 2 for a given CPU time with $N_i = 2$. Case study 2 is the largest in terms of model size among the four case studies. When we allow a CPU time of 0.3 s, the best MILP objective values obtained by **U-SOS1-I** and **B-SOS1-I** are 400956.47 (PG = 0) and 416727.51 (PG = 0.04), respectively. Thus, the MILP objective improves faster for **B-SOS1-I** than **U-SOS1-I**. This again highlights the benefit of bivariate partitioning.

Conclusions

We addressed piecewise linear relaxation of BLPs using a variety of modeling approaches and partitioning strategies. Using four moderate-size process synthesis problems, we presented a detailed numerical comparison of the bivariate versus univariate partitioning schemes. We used uniform placement of grid points for partitioning based on our proof that it results in the least sum of squares of the maximum separations of individual LP relaxations. During the process, we also evaluated the effectiveness of the special ordered set (SOS) formulations versus CC and IC formulations. A formulation with SOS1 construction seems to be the best option for both univariate and bivariate partitioning. Although bivariate partitioning scheme does not seem more attractive than the univariate scheme in solution efficiency, it improves the relaxation quality consistently. Although neither univariate nor bivariate partitioning give guaranteed, better overall computational performance of a GO algorithm, keeping in mind the tradeoff between solution time and relaxation quality, a 2-partition-based bivariate partitioning scheme may be attractive.

Acknowledgments

The authors acknowledge the financial support for this work from the National University of Singapore (NUS) through the research scholarship program and a Cross-Faculty Research Grant (R-279-000-246-123). They also thank Ms. Sangeeta Balram for her useful suggestions in improving some aspects of this manuscript.

Notation

i, j = variable
x_i = variable i
z_{ij} = bilinear product of x_i and x_j
N_i = number of segments into which x_i is partitioned
a_{in} = grid point n defining the partitions
d_i = length of each partition of x_i
Δx_i = global differential variable for x_i
Δz_{ij} = global differential variable for z_{ij}
Δv_{ijn} = bilinear product of μ_{in} and Δx_j
y_i = 1 if $i \notin \Pi$
μ_{in} = 1 if $x_i \geq nd_i$
λ_{in} = 1 if $(n-1)d_i \leq x_i \leq nd_i$
η_{in} = 1 if only ζ_{in} and $\zeta_{i(n+1)}$ are positive
ζ_{in} = SOS2 variable for x_i at segment n
w_{ijn} = bilinear product of ζ_{in} and x_j
θ_{ijnm} = bilinear product of μ_{in} and μ_{jm}
ω_{ijnm} = bilinear product of ζ_{in} and ζ_{jm}
δ_{ijnm} = bilinear product of λ_{in} and λ_{jm}

Literature Cited

1. Meyer CA, Floudas CA. Global optimization of a combinatorially complex generalized pooling problem. *AIChE J*. 2006;52:1027–1037.
2. Karuppiah R, Grossmann IE. Global optimization for the synthesis of integrated water systems in chemical processes. *Comput Chem Eng*. 2006;30:650–673.
3. Takama N, Kuriyama Y, Shiroko K, Umeda T. Optimal water allocation in a petrochemical refinery. *Comput Chem Eng*. 1980;4:251–258.
4. Björk KJ, Westerlund T. Global optimization of heat exchanger network synthesis problems with and without the isothermal mixing assumption. *Comput Chem Eng*. 2002;26:1581–1593.
5. Hasan MMF, Karimi IA, Alfadala HE, Grootjans H. Operational modeling of multi-stream heat exchangers with phase changes. *AIChE J*. 2009;55:150–171.
6. Haverly CA. Studies of the behaviour of recursion for the pooling problem. *ACM SIGMAP Bull*. 1978;25:19–28.
7. Pham V, Laird C, El-Halwagi M. Convex hull discretization approach to the global optimization of pooling problems. *Ind Eng Chem Res*. 2009;48:1973–1979.
8. Reddy PCP, Karimi IA, Srinivasan R. A Novel solution approach for optimizing crude oil operations. *AIChE J*. 2004;50:1177–1197.
9. Li J, Li W, Karimi IA, Srinivasan R. Improving the robustness and efficiency of crude scheduling algorithms. *AIChE J*. 2007;53:2659–2680.
10. Li J, Karimi IA, Srinivasan R. Recipe determination and scheduling of gasoline blending and distribution operations. *AIChE J*. 2009, In Press.
11. Floudas, CA, Pardalos PM, Adjiman CS, Esposito WR, Gumus ZH, Harding ST, Klepeis JL, Meyer CA, Schweiger CA. *Handbook of Test Problems in Local and Global Optimization*. Dordrecht, The Netherlands: Kluwer Academic, 1999.
12. Hasan MMF, Karimi IA. Optimal synthesis of fuel gas networks. Paper Presented at the INFORMS Annual Meeting, Washington, DC, USA, October 12–15, 2008.
13. Floudas CA. *Deterministic Global Optimization: Theory, Methods, and Applications*. Dordrecht, The Netherlands: Kluwer Academic, 2000.
14. GrossmannIE, editor. *Global Optimization in Engineering Design*. Dordrecht, The Netherlands: Kluwer Academic, 1996.
15. Tawarmalani M, Sahinidis NV. *Convexification and Global Optimization in Continuous and Mixed-integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*. Dordrecht, The Netherlands: Kluwer Academic, 2002.
16. FloudasCA, Pardalos PM, editors. *Frontiers in Global Optimization*. Dordrecht, The Netherlands: Kluwer Academic, 2004.
17. Horst R, Tuy H. *Global Optimization: Deterministic Approaches*, 2nd ed. Germany: Springer-Verlag, 1993.
18. Tuy H. *Convex Analysis and Global Optimization*. Dordrecht, The Netherlands: Kluwer Academic, 1998.
19. McCormick GP. Computability of global solutions to factorable nonconvex programs, Part I: Convex underestimating problems. *Math Program*. 1976;10:146–175.
20. Al-Khayyal FA, Falk JE. Jointly constrained biconvex programming. *Math Oper Res*. 1983;8:273–286.
21. Foulds LR, Haughland D, Jornsten K. A bilinear approach to the pooling problem. *Optimization*. 1992;24:165–180.
22. Ben-Tal A, Eiger G, Gershovitz V. Global minimization by reducing duality gap. *Math Program*. 1994;63:193–212.
23. Adhya N, Tawarmalani M, Sahinidis NV. A Lagrangian approach to the pooling problems. *Ind Eng Chem Res*. 1999;38:1956–1972.
24. Quesada I, Grossmann IE. Global optimization of bilinear process networks with multicomponent flows. *Comput Chem Eng*. 1995;19:1219–1242.
25. Bergamini ML, Aguirre PA, Grossmann IE. Logic-based outer approximation for globally optimal synthesis of process networks. *Comput Chem Eng*. 2005;29:1914–1933.
26. Wicaksono DS, Karimi IA. Piecewise MILP under- and overestimators for global optimization of bilinear programs. *AIChE J*. 2008;54:991–1008.
27. Wicaksono DS, Karimi IA. Modeling piecewise under- and overestimators for bilinear process network synthesis via mixed-integer linear programming. Paper Presented at the 18th European Symposium on Computer-Aided Process Engineering-ESCAPE18, Lyon, France, June 1–4, 2008.
28. Bergamini ML, Scenna NJ, Aguirre PA. Global optimal structures of heat exchanger networks by piecewise relaxation. *Ind Eng Chem Res*. 2007;46:1752–1763.
29. Misener R, Gounaris CF, Floudas CA. Global optimization of gas lifting operations: a comparative study of piecewise linear formulations. *Ind Eng Chem Res*. 2009;48:6098–6104.
30. Keha AB, de Farias IR Jr, Nemhauser GL. Models for representing piecewise linear cost functions. *Oper Res Lett*. 2004;32:44–48.
31. Beale ELM, Tomlin JA. Special facilities in a general mathematical programming system for nonconvex problems using ordered sets of variables. In: Lawrence J, editor. *Proceedings of the Fifth International Conference on Operations Research*. London: Tavistock Publications, 1970, pp. 447–454.
32. Gounaris CE, Misener R, Floudas CA. Computational comparison of piecewise-linear relaxations for pooling problems. *Ind Eng Chem Res*. 2009;48:5742–5766.
33. Androulakis IP, Maranas CD, Floudas CA. αBB: a global optimization method for general constrained nonconvex problems. *J Global Optim*. 1995;7:337–363.
34. Padberg MW. Approximating separable nonlinear functions via mixed zero-one programs. *Oper Res Lett*. 2000;27:1–5.
35. Wolsey LA. Mixed integer programming for production planning and scheduling. In Proceedings of Foundations of Computer-Aided Process Operations (FOCAPO2008), Cambridge, Massachusetts, USA, June 29–July 2, 2008.
36. GAMS Development Corporation. *CPLEX, GAMS: the Solver Manuals*. Available at: http://www.gams.com/dd/docs/solvers/cplex.pdf. Accessed 2005.
37. Balas E. Disjunctive programming: properties of the convex hull of feasible points. *Discrete Appl Math*. 1998;89:3–44.
38. Yee TF, Grossmann IE, Kravanja Z. Simultaneous optimization models for heat integration-I. Area and energy targeting and modeling of multi-stream exchangers. *Comput Chem Eng*. 1990;14:1151–1164.
39. Tawarmalani M, Ahmed S, Sahinidis NV. Product disaggregation in global optimization and relaxations of rational programs. *J Global Optim*. 2002;3:281–303.
40. Liberti L, Pantelides CC. An exact reformulation algorithm for large nonconvex NLPs involving bilinear terms. *J Global Optim*. 2006;36:161–189.
41. Liu Y, Karimi IA. Scheduling multistage, multiproduct batch plants with nonidentical parallel units and unlimited intermediate storage. *Chem Eng Sci*. 2007;62:1549–1566.

Appendix A: Maximum departure of *z* from its convex and concave envelopes

The LP relaxation for $z = xy$ with $0 \leq x \leq x^{\mathrm{U}}$, $0 \leq y \leq y^{\mathrm{U}}$ is given by:

$$z \geq 0 \tag{A1}$$

$$z \geq y^{\mathrm{U}}x + x^{\mathrm{U}}y - x^{\mathrm{U}}y^{\mathrm{U}} \tag{A2}$$

$$z \le x^{U} y \tag{A3}$$

$$z \le y^{U} x \tag{A4}$$

The maximum departure of z from its LP relaxation can be obtained by solving the following optimization problem:

$$\min_{x,y,z} \ |xy - z|$$
$$\text{subject to} \quad z - x^{U}y \le 0$$
$$z - y^{U}x \le 0$$
$$y^{U}x + x^{U}y - x^{U}y^{U} - z \le 0$$
$$-z \le 0, -x \le 0, -y \le 0, x - x^{U} \le 0, y - y^{U} \le 0,$$

Consider $\min_{x,y,z}(xy - z)$ first. Let π_1, π_2, π_3, π_4, π_5, π_6, π_7, $\pi_8 \ge 0$ be the Lagrange multipliers for the above inequalities in the order they are mentioned. Because none of $x = 0$, $y = 0$, $x = x^{U}$, and $y = y^{U}$ can represent an optimal solution, we set $\pi_5 = \pi_6 = \pi_7 = \pi_8 = 0$. Then, the Lagrangian (L) and KKT conditions are as follows:

$$L = xy - z + (z - x^{U}y)\pi_1 + (z - y^{U}x)\pi_2 + (y^{U}x + x^{U}y - x^{U}y^{U} - z)\pi_3 - z\pi_4 \tag{A5}$$

$$\pi_3 + \pi_4 = \pi_1 + \pi_2 - 1 \tag{A6}$$

$$x = (\pi_1 - \pi_3)x^{U} \tag{A7}$$

$$y = (\pi_2 - \pi_3)y^{U} \tag{A8}$$

$$(z - x^{U}y)\pi_1 = 0 \tag{A9}$$

$$(z - y^{U}x)\pi_2 = 0 \tag{A10}$$

$$\left(y^{U}x + x^{U}y - x^{U}y^{U} - z\right)\pi_3 = 0 \tag{A11}$$

$$z\pi_4 = 0 \tag{A12}$$

$$x, y > 0, \pi_1, \pi_2, \pi_3, \pi_4 \ge 0, x < x^{U}, y < y^{U} \tag{A13}$$

From Eqs. A6 to A8, we obtain $x = (\pi_4{+}1{-}\pi_2)x^{U}$ and $y = (\pi_4{+}1{-}\pi_1)y^{U}$. These imply $\pi_1 > 0$ and $\pi_2 > 0$, because $x < x^{U}$ and $y < y^{U}$. Using these, we get $z = yx^{U} = xy^{U}$ or $z > 0$ from Eqs. A9, A10, and A13. This gives us $\pi_3 = 0$ and $\pi_4 = 0$ from Eqs. A11 to A12. Therefore, $\pi_1 = \pi_2$ from Eqs. A7 to A8. This also implies $\pi_1 = \pi_2 = 1/2$ from Eq. A6. Thus, $x = x^{U}/2$, $y = y^{U}/2$, $z = x^{U}y^{U}/2$, and $\min_{x,y,z}(xy - z) = -x^{U}y^{U}/4$. Similarly, we can show that $\min_{x,y,z}(xy - z) = -x^{U}y^{U}/4$. For this case, $x = x^{U}/2$, $y = y^{U}/2$, and $z = 0$. Therefore, $\min_{x,y,z} \ |xy - z|$ is $x^{U}y^{U}/4$ and occurs at $x = x^{U}/2$ and $y = y^{U}/2$.

Appendix B: Optimal Segment Lengths

Univariate partitioning

Let x in an arbitrary bilinear product $z = xy$ be partitioned into N segments ($n = 1, 2, \ldots, N$) of lengths d_n. From Appendix A, $d_n/4$ is the maximum departure of $z = xy$ from its LP relaxation in partition n. To obtain the optimal segment lengths, we minimize the sum of squares of all departures as follows:

$$\text{Minimize} \ \sum_{n=1}^{N} \frac{d_n^2}{16} \quad \text{subject to} \ \sum_{n=1}^{N} d_n = 1$$

Let $d_n = u_n^2$, and α be the Lagrange multiplier for the equality constraint. The KKT conditions of the above gives us $d_n = -8\alpha$. Substituting in $\sum_{n=1}^{N} d_n = 1$ gives us $8N\alpha + 1 = 0$ and $d_n = 1/N$. Thus, uniform placement seems to the best scheme for univariate partitioning.

Bivariate partitioning

Let x have N and y have M segments for $z = xy$ with lengths d_{xn} ($n = 1, 2, \ldots, N$) and d_{ym} ($m = 1, 2, \ldots, M$). Then, for the bivariate case, we have,

$$\text{Minimize} \ \sum_{n=1}^{N}\sum_{m=1}^{M} \frac{d_{xn}^2 d_{ym}^2}{16}$$
$$\text{subject to} \ \sum_{n=1}^{N} d_{xn} = 1 \quad \text{and} \quad \sum_{m=1}^{M} d_{ym} = 1$$

If α and β are the Lagrange multipliers for the two equalities, $d_{xn} = u_n^2$, and $d_{ym} = v_m^2$, the KKT conditions give us $d_{xn} = -2\alpha$ and $d_{ym} = -2\beta$. Substituting back in the two equalities gives us $d_{xn} = 1/N$ and $d_{ym} = 1/M$. Again, uniform placement is the best choice.

Appendix C: MIBLP model for HENS in Case Study 1

Let h, c, and k denote hot stream, cold stream, and stage, respectively. Also, let HU, CU, K, IN, and OUT represent hot utility, cold utility, total number of stages, inlet, and outlet, respectively. The HENS model involves the following parameters and variables.

Parameters

CF_{hc}, $CF_{h,CU}$, $CF_{c,HU}$ = fixed costs for heat exchangers (HE), coolers, and heaters
CCU, CHU = per unit cost of cold, hot utility
C_{hc}, $C_{h,CU}$, $C_{c,HU}$ = area cost coefficients
U_{hc}, $U_{h,CU}$, $U_{c,HU}$ = overall heat transfer coefficients
$T_{h,IN}$, $T_{h,OUT}$ = inlet and outlet temperatures of hot stream h
$T_{c,IN}$, $T_{c,OUT}$ = inlet and outlet temperatures of cold stream c
$T_{HU,IN}$, $T_{HU,OUT}$ = inlet and outlet temperatures of hot utility
$T_{CU,IN}$, $T_{CU,OUT}$ = inlet and outlet temperatures of cold utility
F_i, F_j = heat capacity flow rates
δ = minimum approach temperature
Ω = upper bound on heat transfer
Γ = upper bound on temperature difference

Binary variables

z_{hck} = 1 if hot stream h contacts cold stream c at stage k
zcu_h = 1 if hot stream h contacts cold utility
zhu_c = 1 if cold stream c contacts hot utility

Continuous variables

q_{hck} = heat duty of the HE corresponding to match (h, c, k)
$\mathrm{qcu_h}$ = heat duty of the cooler corresponding to hot stream h
$\mathrm{qhu_c}$ = heat duty of the heater corresponding to cold stream c
A_{hck} = area of the HE corresponding to match (h, c, k)
$\mathrm{Acu_h}$ = area of the cooler corresponding to hot stream h
$\mathrm{Ahu_c}$ = area of the heater corresponding to cold stream c
$\mathrm{dth_{hck}}$ = temperature approach in the hot end of HE (h, c, k)
$\mathrm{dtc_{hck}}$ = temperature approach in the cold end of HE (h, c, k)
$\mathrm{dtcu_h}$ = temperature approach in the hot end of cooler for hot stream h
$\mathrm{dthu_c}$ = temperature approach in the cold end of heater for cold stream c
t_{hk} = temperature of hot stream h at the hot end of stage k
t_{ck} = temperature of cold stream c at the hot end of stage k
$\mathrm{th_{hck}}$ = temperature of part of the hot stream h after HE (h, c, k)
$\mathrm{tc_{hck}}$ = temperature of part of the cold stream c after HE (h, c, k)
$\mathrm{fh_{hck}}$ = fraction of the flow of hot stream h in HE (h, c, k)
$\mathrm{fc_{hck}}$ = fraction of the flow of cold stream c in HE (h, c, k)

Unless stated otherwise, all indices assume the full ranges of their valid values in all the constraints. The HENS model is as follows.

Objective function:

$$\text{Minimize} \quad \sum_{\mathrm{h}}\sum_{\mathrm{c}}\sum_{\mathrm{k}} \mathrm{CF_{hc}} z_{\mathrm{hck}} + \sum_{\mathrm{h}} \mathrm{CF_{h,CU}zcu_h} + \sum_{\mathrm{c}} \mathrm{CF_{c,HU}zhu_c} + \sum_{\mathrm{h}} \mathrm{CCUqcu_h} + \sum_{\mathrm{c}} \mathrm{CHUqhu_c} + \sum_{\mathrm{h}}\sum_{\mathrm{c}}\sum_{\mathrm{k}} C_{\mathrm{hc}} A_{\mathrm{hck}} + \sum_{\mathrm{h}} C_{\mathrm{h,CU}} \mathrm{Acu_h} + \sum_{\mathrm{c}} C_{\mathrm{c,HU}} \mathrm{Ahu_c} \tag{C1}$$

Stream splitting:

$$\sum_{\mathrm{c}} \mathrm{fh_{hck}} = \sum_{\mathrm{h}} \mathrm{fc_{hck}} = 1 \tag{C2}$$

Overall energy balance for each stream:

$$\sum_{\mathrm{c}}\sum_{\mathrm{k}} q_{\mathrm{hck}} + \mathrm{qcu_h} = F_{\mathrm{h}}\left(T_{\mathrm{h,IN}} - T_{\mathrm{h,OUT}}\right) \tag{C3a}$$

$$\sum_{\mathrm{h}}\sum_{\mathrm{k}} q_{\mathrm{hck}} + \mathrm{qhu_c} = F_{\mathrm{c}}\left(T_{\mathrm{c,OUT}} - T_{\mathrm{c,IN}}\right) \tag{C3b}$$

Energy balance at each stage:

$$\sum_{\mathrm{c}} q_{\mathrm{hck}} = F_{\mathrm{h}}\left(t_{\mathrm{hk}} - t_{\mathrm{h(k+1)}}\right) \tag{C4a}$$

$$\sum_{\mathrm{h}} q_{\mathrm{hck}} = F_{\mathrm{c}}\left(t_{\mathrm{ck}} - t_{\mathrm{c(k+1)}}\right) \tag{C4b}$$

Energy balance for each heat exchanger

$$q_{\mathrm{hck}} = \mathrm{fh_{hck}} F_{\mathrm{h}}(t_{\mathrm{hk}} - \mathrm{th_{hck}}) = \mathrm{fc_{hck}} F_{\mathrm{c}}\left(\mathrm{tc_{hck}} - t_{\mathrm{c(k+1)}}\right) \tag{C5}$$

Hot and cold utility balances:

$$\mathrm{qcu_h} = F_{\mathrm{h}}\left(t_{\mathrm{h(K+1)}} - T_{\mathrm{h,OUT}}\right) \tag{C6a}$$

$$\mathrm{qhu_c} = F_{\mathrm{c}}\left(T_{\mathrm{c,OUT}} - t_{\mathrm{c1}}\right) \tag{C6b}$$

Fix inlet temperatures:

$$t_{\mathrm{h1}} = T_{\mathrm{h,IN}} \tag{C7a}$$

$$t_{\mathrm{c(K+1)}} = T_{\mathrm{c,IN}} \tag{C7b}$$

Monotonic decrease in temperatures:

$$t_{\mathrm{hk}} \geq t_{\mathrm{h(k+1)}} \geq T_{\mathrm{h,OUT}} \tag{C8}$$

$$T_{\mathrm{c,OUT}} \geq t_{\mathrm{ck}} \geq t_{\mathrm{c(k+1)}} \tag{C9}$$

$$t_{\mathrm{hk}} \geq th_{\mathrm{hck}} \tag{C10a}$$

$$t_{\mathrm{c(k+1)}} \leq \mathrm{tc_{hck}} \tag{C10b}$$

Logical constraints:

$$q_{\mathrm{hck}} \leq \Omega z_{\mathrm{hck}} \tag{C11a}$$

$$\mathrm{qcu_h} \leq \Omega \mathrm{zcu_h} \tag{C11b}$$

$$\mathrm{qhu_c} \leq \Omega \mathrm{zhu_c} \tag{C11c}$$

Approach temperatures:

$$\mathrm{dth_{hck}} \leq t_{\mathrm{hk}} - \mathrm{tc_{hck}} + \Gamma(1 - z_{\mathrm{hck}}) \tag{C12a}$$

$$\mathrm{dtc_{hck}} \leq \mathrm{th_{hck}} - t_{\mathrm{c(k+1)}} + \Gamma(1 - z_{\mathrm{hck}}) \tag{C12b}$$

$$\mathrm{dtcu_h} \leq t_{\mathrm{h(K+1)}} - T_{\mathrm{CU,OUT}} + \Gamma(1 - \mathrm{zcu_h}) \tag{C13a}$$

$$\mathrm{dthu_c} \leq T_{\mathrm{HU,OUT}} - t_{\mathrm{c1}} + \Gamma(1 - \mathrm{zhu_c}) \tag{C13b}$$

Heat transfer equations:

$$q_{\mathrm{hck}} = U_{\mathrm{hc}} A_{\mathrm{hck}} \left(\frac{\mathrm{dth_{hck}} + \mathrm{dtc_{hck}}}{2}\right) \tag{C14a}$$

$$\mathrm{qcu_h} = U_{\mathrm{h,CU}} \mathrm{Acu_h} \left(\frac{\mathrm{dtc_h} + T_{\mathrm{h,OUT}} - T_{\mathrm{CU,IN}}}{2}\right) \tag{C14b}$$

$$\mathrm{qhu_c} = U_{\mathrm{c,HU}} \mathrm{Ahu_c} \left(\frac{\mathrm{dthu_c} + T_{\mathrm{HU,IN}} - T_{\mathrm{c,OUT}}}{2}\right) \tag{C14c}$$

Variable bounds:

$$0 \leq \mathrm{fh_{hck}} \leq 1, 0 \leq \mathrm{fc_{hck}} \leq 1, \mathrm{dth_{hck}} \geq \delta, \mathrm{dtc_{hck}} \geq \delta, \mathrm{dthu_c} \geq \delta, \mathrm{dtcu_h} \geq \delta, T_{\mathrm{h,OUT}} \leq t_{\mathrm{hk}} \leq T_{\mathrm{h,IN}}, T_{\mathrm{c,IN}} \leq t_{\mathrm{ck}} \leq T_{\mathrm{c,OUT}}, T_{\mathrm{h,OUT}} \leq \mathrm{th_{hck}} \leq T_{\mathrm{h,IN}}, T_{\mathrm{c,IN}} \leq \mathrm{tc_{hck}} \leq T_{\mathrm{c,OUT}}, 0 \leq q_{\mathrm{hck}} \leq \min\left[F_{\mathrm{h}}\left(T_{\mathrm{h,IN}} - T_{\mathrm{h,OUT}}\right), F_{\mathrm{c}}\left(T_{\mathrm{c,OUT}} - T_{\mathrm{c,IN}}\right)\right], 0 \leq \mathrm{qcu_h} \leq F_{\mathrm{h}}\left(T_{\mathrm{h,IN}} - T_{\mathrm{h,OUT}}\right), \text{ and } 0 \leq \mathrm{qhu_c} \leq F_{\mathrm{c}}\left(T_{\mathrm{c,OUT}} - T_{\mathrm{c,IN}}\right).$$

We use a minimum approach of 10 K, $\mathbf{\Omega} = 10^6$, and $\mathbf{\Gamma} = 10^3$. The fixed costs of heat exchangers, heaters, and coolers are US$15000. The area cost coefficients are taken as 30 for all exchangers and coolers, and 60 for heaters. The overall heat transfer coefficients are taken as 0.0857, 0.06, 0.067, 0.05, 0.1154, .0833, 0.18182, and 0.09524 for matches H1-C1, H1-C2, H2-C1, H2-C2, H1-cooler, H2-cooler, C1-heater, and C2-heater, respectively. Costs of unit hot and cold utilities are US$110 and US$10, respectively.

Appendix D: MIBLP model of the pooling problem from MF in Case Study 2

Let s, c, e, and t denote source, quality, sink, and plant, respectively. Let S, C, E, and T denote the set of sources, qualities, sinks, and plants, respectively. MF model involves the following parameters and variables.

Parameters

f_s^{source} = flow rate of source s
q_{cs}^{source} = value of quality c in source s
$q_{ce}^{\max}$ = maximum allowable value of quality c in sink e
r_{ct} = removal ratio of quality c in plant t
c_{se}^a = cost per unit flow from source s to sink e
c_{te}^b = cost per unit flow from plant t to sink e
$c_{tt'}^c$ = cost per unit flow from plant t to plant t'
c_{st}^d = cost per unit flow from source s to plant t
c_t^e = cost per unit flow through plant t
c_{se}^{ya} = fixed cost of pipeline from source s to sink e
c_{te}^{yb} = fixed cost of pipeline from plant t to sink e
$c_{tt'}^{yc}$ = fixed cost of pipeline from plant t to plant t'
c_{st}^{yd} = fixed cost of pipeline from source s to plant t
c_t^{ye} = fixed cost of plant t

Binary variables

y_{se}^a = 1 if stream connecting source s to sink e is selected
y_{te}^b = 1 if stream connecting plant t to sink e is selected
$y_{tt'}^c$ = 1 if directed stream connecting plant t to plant t' is selected
y_{st}^d = 1 if stream connecting source s to plant t is selected
y_t^e = 1 if plant t is selected

Continuous variables

a_{se} = flow rate of stream connecting source s to sink e
b_{te} = flow rate of stream connecting plant t to sink e
$c_{tt'}$ = flow rate of directed stream connecting plant t to plant t'
d_{st} = flow rate of stream connecting source s to plant t
e_t = flow rate of plant t effluent

Objective function:

$$\begin{aligned}\text{Minimize} \quad & \sum_{s\in S} c_s^a\left(f_s^{\text{source}} - \sum_{t\in T} d_{st}\right) + \sum_{t\in T}\sum_{s\in S} c_t^b d_{st} \\ & + \sum_{t\in T}\left(\sum_{t'\in T\setminus\{t\}} c_t^b(c_{t't} - c_{tt'}) + \sum_{t'\in T\setminus\{t\}} \left(c_{tt'}^c + c_{t'}^e\right)c_{tt'}\right) \\ & + \sum_{s\in S}\sum_{t\in T}\left(c_{st}^d + c_t^e\right)d_{st} + \sum_{s\in S} c_s^{ay} y_s^a + \sum_{t\in T} c_t^{by} y_t^b \\ & + \sum_{t\in T}\sum_{t'\in T\setminus\{t\}} c_{tt'}^{by} y_{tt'}^c + \sum_{s\in S}\sum_{t\in T} c_{st}^d y_{st}^d + \sum_{t\in T} c_t^{ey} y_t^e \end{aligned} \tag{D1}$$

Constraints:

$$f_s^{\text{source}} - \sum_{t\in T} d_{st} - y_s^a \bar{a}_s \le 0 \qquad s\in S \tag{D2}$$

$$\sum_{t'\in T\setminus\{t\}} c_{t't} - \sum_{t'\in T\setminus\{t\}} c_{tt'} + \sum_{s\in S} d_{st} - y_t^b \bar{b}_t \le 0 \qquad t\in T \tag{D3}$$

$$c_{tt'} - y_{tt'}^c \bar{c}_{tt'} \le 0 \qquad t\in T, t'\in T\setminus\{t\} \tag{D4}$$

$$d_{st} - y_{st}^d \bar{d}_{st} \le 0 \qquad s\in S, t\in T \tag{D5}$$

$$y_s^a \underline{a}_s - f_s^{\text{source}} + \sum_{t\in T} d_{st} \le 0 \qquad s\in S \tag{D6}$$

$$y_t^b \underline{b}_t - \sum_{t'\in T\setminus\{t\}} c_{t't} + \sum_{t'\in T\setminus\{t\}} c_{tt'} - \sum_{s\in S} d_{st} \le 0 \qquad t\in T \tag{D7}$$

$$y_{tt'}^c, \underline{c}_{tt'} - c_{tt'} \le 0 \qquad t\in T, t'\in T\setminus\{t\} \tag{D8}$$

$$y_{st}^d \underline{d}_{st} - d_{st} \le 0 \qquad s\in S, t\in T \tag{D9}$$

$$\sum_{s\in S} d_{st} + \sum_{t'\in T\setminus\{t\}} c_{t't} - y_t^e \bar{e}_t \le 0 \qquad t\in T \tag{D10}$$

$$-\sum_{s\in S} d_{st} - \sum_{t'\in T\setminus\{t\}} c_{t't} + y_t^e \underline{e}_t \le 0 \qquad t\in T \tag{D11}$$

$$y_{tt'}^c + y_{t't}^c \le 1 \qquad t\in T, t'\in T\setminus\{t\} \tag{D12}$$

$$\begin{aligned} & \sum_{s\in S} q_{ct} d_{st} + \sum_{t'\in T\setminus\{t\}} q_{ct} c_{t't} \\ & = (1 - r_{ct})\left(\sum_{t'\in T\setminus\{t\}} q_{ct'} c_{t't} + \sum_{s\in S} q_{cs}^{\text{source}} d_{st}\right) \quad c\in C, t\in T \end{aligned} \tag{D13}$$

$$\begin{aligned} & \sum_{s\in S} f_s^{\text{source}}\left(q_{cs}^{\text{source}} - q_c^{\max}\right) + \sum_{s\in S}\sum_{t\in T} d_{st}\left(-q_{cs}^{\text{source}} + q_{ct}\right) \\ & + \sum_{t\in T}\sum_{t'\in T} d_{st}\left(q_{ct} - q_c^{\max}\right)(c_{t't} - c_{tt'}) \le 0 \quad c\in C, t\in T \end{aligned} \tag{D14}$$

Variable bounds:

$$0 \le a_{se} \le \bar{a}_{se},\ 0 \le b_{te} \le \bar{b}_{te},\ 0 \le c_{tt'} \le \bar{c}_{tt'},\ 0 \le d_{st} \le \bar{d}_{st},\ \text{and } 0 \le q_{ct} \le \bar{q}_{ct}.$$

Appendix E: BLP model from KG in Case Study 3

We use the following BLP model from KG in case study 3.

Sets and indices

i, k = stream indices
j = contaminant
m = mixer
m_{in} = set of inlet streams into mixer m
m_{out} = outlet stream from mixer m
MU = set of mixers
J = set of contaminants
n = interval
p = process unit
p_{in} = inlet stream into process unit p
p_{out} = outlet stream from process unit p
PU = set of process units
r = treatment technology
s = splitter
s_{in} = inlet stream into splitter s
s_{out} = set of outlet streams from splitter s
SU = set of splitters
t = treatment unit
t_{in} = inlet stream into treatment unit t
t_{out} = outlet stream from treatment unit t
TU = set of treatment units

Parameters

AR = annualized factor for investment on treatment units
C_{FW} = cost of freshwater
C_j^{iL} = lower bound on concentration of contaminant j in stream i
C_j^{iU} = upper bound on concentration of contaminant j in stream i
C_j^{riL} = lower bound on concentration of contaminant j in input/output stream i of treatment technology r
C_j^{riU} = upper bound on concentration of contaminant j in input/output stream i of treatment technology r
F^{iL} = lower bound on flow in stream i
F^{iU} = upper bound on flow in stream i
F^{riL} = lower bound on flow in input/output stream i of treatment technology r
F^{riU} = upper bound on flow in input/output stream i of treatment technology r
H = hours of plant operation per annum
IC^t = investment cost coefficient for treatment unit t
L_j^p = load of contaminant j inside process unit p
N = total number of intervals used for partitioning each flow
OC^t = operating cost coefficient for treatment unit t
P^p = flow demand in process unit p
α = cost function exponent $(0 < \alpha \leq 1)$
$\beta_j^t = 1 - \{(\text{removal ratio for contaminant } j \text{ in unit } t \text{ (in \%)})/100\}$
$\beta_j^{rt} = 1 - \{(\text{removal ratio for contaminant } j \text{ in unit } t \text{ using technology } r \text{ (in \%)})/100\}$
γ^{rt} = investment cost coefficient for treatment unit t using technology r
δ_j = maximum concentration of contaminant j allowed in discharge
ζ_j = maximum flow of contaminant j allowed in discharge
Θ^{rt} = operating cost coefficient for treatment unit t using technology r

Continuous variables

C_j^i = concentration of contaminant j in stream i
f_j^i = flow of contaminant j in stream i
f_j^{out} = flow of contaminant j in the outlet stream to the environment
F^i = flow rate of stream i
FW = freshwater intake into the system
INV^t = investment cost for treatment unit t
OP^t = operating cost for treatment unit t

Binary variables

w_{rn}^t = 1 if flow through the rth treatment technology for treatment unit t lies in the nth interval
y_{rt} = 1 if rth treatment technology is chosen for treatment unit t
λ_n^i = 1 if the flow variable F^i takes a value in the nth interval

Objective function:

$$\text{Minimize} \quad \sum_{i \in s1_{\text{out}}} F^i + \sum_{\substack{t \in \text{TU} \\ i \in t_{\text{out}}}} F^i \tag{E1}$$

Mixer units:

$$F^k = \sum_{i \in m_{\text{in}}} F^i \qquad m \in \text{MU}, k \in m_{\text{out}} \tag{E2}$$

$$F^k C_j^k = \sum_{i \in m_{\text{in}}} F^i C_j^i \qquad j \in J, m \in \text{MU}, k \in m_{\text{out}} \tag{E3}$$

Splitter units:

$$F^k = \sum_{i \in s_{\text{out}}} F^i \qquad m \in \text{SU}, k \in s_{\text{in}} \tag{E4}$$

$$C_j^i = C_j^k \qquad j \in J, s \in \text{SU}, i \in s_{\text{out}}, k \in s_{\text{in}} \tag{E5}$$

Process units:

$$P^p C_j^i + 10^3 L_j^p = P^p C_j^k \qquad j \in J, p \in \text{PU}, i \in p_{\text{in}}, k \in p_{\text{out}} \tag{E6}$$

Treatment units:

$$F^k = F^i \qquad t \in \text{TU}, i \in t_{\text{out}}, k \in t_{\text{in}} \tag{E7}$$

$$C_j^i = \beta_j^t C_j^k \qquad j \in J, t \in \text{TU}, i \in t_{\text{out}}, k \in t_{\text{in}} \tag{E8}$$

Bound strengthening cut:

$$\sum_{p \in \text{PU}} 10^3 L_j^p = \sum_{\substack{t \in \text{TU} \\ i \in t_{\text{in}}}} \left(1 - \beta_j^t\right) f_j^k + f_j^{\text{out}} \qquad j \in J \tag{E9}$$

Also, note that $F^k = F^i = P^p$ for $p \in \text{PU}$, $i \in p_{\text{in}}$, $k \in p_{\text{out}}$. We also fix the known flows.

Manuscript received Apr. 24, 2009, revision received July 18, 2009, and final revision received Oct. 1, 2009.