I&EC research
Industrial & Engineering Chemistry Research

Article

pubs.acs.org/IECR

Global Optimization of the Stage-wise Superstructure Model for Heat Exchanger Networks

Débora C. Faria, Sung Young Kim, and Miguel J. Bagajewicz*

University of Oklahoma 100 E. Boyd Street, Room T335 Norman, Oklahoma 73019, United States

ABSTRACT: We modify our recently introduced bound contraction methodology for global optimization of bilinear MINLP models[1] to solve the stage-wise superstructure model for heat exchanger networks.[2] Because the problem has nonlinearities that are not bilinear or quadratic instead of converting the problem into a fully quadratic one, which would add much more variables and equations, we use a different relaxation approach that takes advantage of monotonicity. Aside from the original bound contraction method, we also use branch and bound combined with bound contraction at each node.

1. INTRODUCTION

Among the superstructure-based models for a heat exchanger network (HEN) design, the most popular one is the stage-wise superstructure approach due to Yee and Grossmann.[2] Even in its simplest form (isothermal mixing), this mixed integer nonlinear programming (MINLP) problem is a challenge to local solvers, not even rendering local minima at times. Because of this, we resort to the use of a robust global optimization method; that is, one that guarantees answers without presenting numerical failures.

There are several global optimization methods, some even commercially available like BARON,[3] COCOS,[4] GlobSol,[5] ICOS,[6] LGO,[7] LINGO,[8] OQNLP,[9] Premium Solver,[10] and others that are well-known like the αBB,[11] GLoMIQO,[12] Antigone,[13] SCIP,[14,15] and Couenne.[16] Many are described in several books[17−21] and recent paper reviews.[22−25]

Some of the above methods as well as others, apply different approaches for global optimization: Lagrangean-based approaches,[26−29] disjunctive programming-based methods,[30] cutting plane methods,[31] interval arithmetic,[32−37] branch-and-reduce,[38] GMIN-αBB and SMIN-αBB,[39] spatial branch and bound,[40] branch and cut[41] and branch and contract,[42,43] decomposition through generalized benders decomposition[44] or outer approximation,[45,46] both restricted to a certain class of problems[47,48] as well as others.[49] The list is not exhaustive.

Bilinear terms abound in process engineering because they represent products of flow rates and concentrations/temperatures in component and energy balances, to mention just the most common ones. In addition, univariate concave terms show up typically in cost functions. In a previous work (Faria and Bagajewicz[1,50−52]) we address problems that contain these bilinear and univariate concave terms. Faria and Bagajewicz[1,50−52] use some previously proposed partitioning methods to obtain lower bounds in addition to some reformulation techniques: those from Sherali and Alamedinne;[53] Meyer and Floudas;[54] Misener and Floudas;[55−59] Gounaris and co-workers;[60] Karuppiah and Grossmann;[61,62] Bergamini and co-workers;[63,64] Wicaksono and Karimi;[65] Hasan and Karimi;[66] and Pham and co-workers.[67] Some of these techniques were incorporated in commercial software that is based on the concept of range partitioning of variables in bilinear terms (Antigone, Couenne, etc.).

In our work we use a methodology based on the aforementioned partitioning of portions of the domain to obtain lower bounds, but departing from other efforts we use a new bound contraction procedure.[1,50−52] The procedure presented differs from most of the previous approaches based on overall lower and upper bounds (OLB/OUB) schemes in that it does not use a branch and bound methodology. Instead, after partitioning the intervals between lower and upper bounds of certain variables into several subintervals, a bound contraction procedure based on the feasibility of the lower bound model is used in an interval elimination strategy. The novelty the previous researches introduced is feasibility-based and objective function value-based bound contraction.

Our additional contribution in this paper is the extension of the method used to underestimate bilinear terms only and bound contract discretized variables developed by Faria and Bagajewicz[1,50−52] to the underestimation of terms containing nonconvex monotone functions. Instead of introducing several new variables to convert the problem into a quadratic one, we use a different relaxation of monotone functions that does not require reformulation and addition of new variables beyond the integers needed for discretization. We also explore the use of branch and bound with our bound contraction procedure at each node.

The paper is organized as follows: We make a brief review of the bound contraction procedure and the newly proposed B&B procedure with bound contraction at each node. We then discuss our relaxation procedure, and finally, examples are presented.

2. GLOBAL OPTIMIZATION STRATEGY

The global optimization strategy we use was developed by Faria and Bagajewicz.[1,50−52] It relies on partitioning variables to build a linear (or convex) lower bound. The algorithm works as follows:

1. The relaxed model (lower bound) is solved. If there is no solution, the problem is infeasible. If not, the objective value found is the first lower bound.

Received: August 13, 2014
Revised: December 19, 2014
Accepted: December 23, 2014

DOI: 10.1021/ie5032315
Ind. Eng. Chem. Res. XXXX, XXX, XXX−XXX

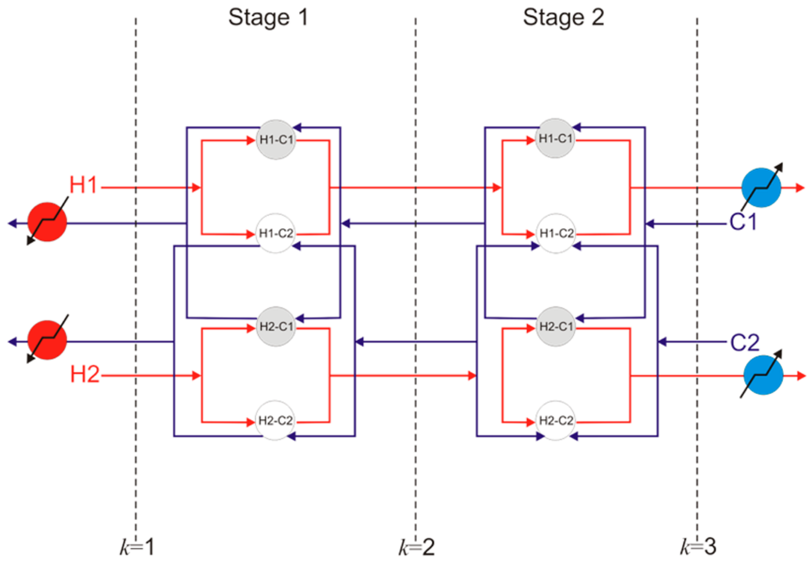

Figure 1. Stage-wise superstructure.

2. The original model is solved using any available MINLP code (in our case, we use DICOPT) using the values of the variables obtained from the lower bound as initial points. This helps in obtaining meaningful upper bounds. We keep the objective as the first upper bound.
3. The bound contraction method is used and as a result a new set of bounds for the variables are obtained and the lower bound is updated. We discuss the bound contraction later.
4. The upper bound is updated and the procedure continues until the gap between lower and upper bound is small enough.

Regarding the number of partitions (see Faria and Bagajewicz[1,50−52]), the strategy in this paper is to pick as few as possible and when the bound contraction does not progress, then increase it slowly.

3. HEN MODEL

Zamora and Grossmann[42] proposed a global optimization algorithm to rigorously optimize the stage-wise superstructure approach due to Yee and Grossmann,[2] under the simplifying assumptions of linear area cost functions and no stream splitting. The approach relies on the use of convex underestimators for the heat transfer area. Later, the approach was extended to account for the nonlinear area cost functions.[68] Adjiman and co-workers solved the HEN synthesis problem under the assumption of linear area cost functions, using the SMIN-αBB algorithm.[69] Björk and Westerlund covered stage-wise models and solved HEN synthesis problems to global optimality under isothermal and nonisothermal mixing assumptions.[70] The most recent attempt to solve the Synheat model to global optimum is the one proposed by Bergamini and co-workers.[71] Their strategy is based on an outer approximation methodology, employing piece-wise underestimators of nonconvex terms and physical insights.

The stage-wise superstructure of the model enables both exchangers in parallel, as well as in series. A two stage superstructure for two hot and two cold streams is presented in Figure 1.

Heat can be transferred between each pair of hot and cold streams in each stage, and if the stream splits in a stage it is remixed isothermally before entering the next stage. The isothermal mixing assumption eliminates the requirement for nonlinear heat balances around heat exchangers as well as nonlinear heat mixing equations. In the original formulation by Yee and Grossmann[2] the feasible space is defined by strictly linear constraints. Nonetheless, the model is nonconvex due to the presence of nonlinear, nonconvex terms in the objective function related to area costs.

The nonconvex MINLP model presented in this work differs slightly from the one reported by Yee and Grossmann.[2] The differences are as follows:

- The areas of heat exchangers are used explicitly in the objective function (The original Synheat model uses the ratio of the heat transferred to the log mean temperature difference).
- The area costs are assumed to be linearly dependent on the areas, thus making the objective function linear.
- Because the areas of heat exchangers are explicitly defined in the objective function, new constraints to calculate them are incorporated.

Although the second assumption can be conceptually challenged, Barbaro and Bagajewicz argued that linear approximations of cost equations can be justified by the fact that they already carry an inherent uncertainty.[72]

We now present the model equations. The reader is referred to the paper by Yee and Grossmann[2] for explanations of each equation. We introduce explanations about what is new.

Objective function:

$$\begin{aligned}\min & \sum_{i\in HP}\sum_{j\in CP}\sum_{k\in ST} C_{HEi,j} z_{i,j,k} + \sum_{i\in HP} C_{HEi} z_{CUi} \\ & + \sum_{j\in CP} C_{HEj} z_{HUj} + \sum_{i\in HP} C_{CU} q_{CUi} + \sum_{j\in CP} C_{HU} q_{HUj} \\ & + \sum_{i\in HP}\sum_{j\in CP}\sum_{k\in ST} C_{Ai,j} A_{i,j,k} + \sum_{i\in HP} C_{Ai} A_{CUi} \\ & + \sum_{j\in CP} C_{Aj} A_{HUj}\end{aligned} \tag{1}$$

where $z_{i,j,k}$, z_{CUi}, and z_{HUj} are binary variables to denote that existence of heat exchangers, cooling utilities, and heating

DOI: 10.1021/ie5032315
Ind. Eng. Chem. Res. XXXX, XXX, XXX−XXX

utilities, respectively. As described above, the area cost has been linearized, although concave terms can be used.

Overall heat balance for each stream:

$$(T^{H}_{INi} - T^{H}_{OUTi})CF^{H}_{i} = \sum_{k\in ST}\sum_{j\in CP} q_{i,j,k} + q_{CUi} \quad i \in HP \tag{2}$$

$$(T^{C}_{OUTj} - T^{C}_{INj})CF^{C}_{j} = \sum_{k\in ST}\sum_{i\in HP} q_{i,j,k} + q_{HUj} \quad j \in CP \tag{3}$$

Stage heat balance:

$$(T^{H}_{i,k} - T^{H}_{i,k+1})CF^{H}_{i} = \sum_{j\in CP} q_{i,j,k} \quad k \in ST, i \in HP \tag{4}$$

$$(T^{C}_{j,k} - T^{C}_{j,k+1})CF^{C}_{j} = \sum_{i\in HP} q_{i,j,k} \quad k \in ST, j \in CP \tag{5}$$

Superstructure inlet temperatures:

$$T^{H}_{INi} = T^{H}_{i,1} \quad i \in HP \tag{6}$$

$$T^{C}_{INj} = T^{C}_{j,NOK+1} \quad j \in CP \tag{7}$$

Feasibility of temperatures (monotonic decrease in temperatures):

$$T^{H}_{i,k} \geq T^{H}_{i,k+1} \quad k \in ST, i \in HP \tag{8}$$

$$T^{H}_{i,NOK+1} \geq T^{H}_{OUTi} \quad i \in HP \tag{9}$$

$$T^{C}_{j,k} \geq T^{C}_{j,k+1} \quad k \in ST, j \in CP \tag{10}$$

$$T^{C}_{j,1} \leq T^{C}_{OUTj} \quad j \in CP \tag{11}$$

Hot and cold utility load:

$$q_{CUi} = CF^{H}_{i}(T^{H}_{i,NOK+1} - T^{H}_{OUTi}) \quad i \in HP \tag{12}$$

$$q_{HUj} = CF^{C}_{j}(T^{C}_{OUTj} - T^{C}_{j,1}) \quad j \in CP \tag{13}$$

Approach temperatures:

$$\Delta T_{i,j,k} \leq T^{H}_{i,k} - T^{C}_{j,k} + \Gamma(1 - z_{i,j,k}) \quad i \in HP, j \in CP, k \in ST \tag{14}$$

$$\Delta T_{i,j,k+1} \leq T^{H}_{i,k+1} - T^{C}_{j,k+1} + \Gamma(1 - z_{i,j,k}) \quad i \in HP, j \in CP, k \in ST \tag{15}$$

$$\Delta T_{CUi} \leq T^{H}_{i,NOK+1} - T_{OUT,CU} + \Gamma(1 - z_{CUi}) \quad i \in HP \tag{16}$$

$$\Delta T_{HUj} \leq T_{OUT,HU} - T^{C}_{j,1} + \Gamma(1 - z_{HUj}) \quad j \in CP \tag{17}$$

Minimum approach temperature (lower bounds):

$$\Delta T_{i,j,k} \geq EMAT \quad i \in HP, j \in CP, k \in ST \tag{18}$$

$$\Delta T_{CUi} \geq EMAT \quad i \in HP \tag{19}$$

$$\Delta T_{HUj} \geq EMAT \quad j \in CP \tag{20}$$

Logical constraints:

$$q_{i,j,k} - \Omega z_{i,j,k} \leq 0 \quad i \in HP, j \in CP, k \in ST \tag{21}$$

$$q_{CUi} - \Omega z_{CUi} \leq 0 \quad i \in HP \tag{22}$$

$$q_{HUj} - \Omega z_{HUj} \leq 0 \quad j \in CP \tag{23}$$

Area calculations using Chen's (1987) approximation for the logarithmic mean temperature difference:

$$q_{i,j,k} - A_{i,j,k}U_{i,j}\sqrt[3]{\Delta T_{i,j,k}\Delta T_{i,j,k+1}\frac{(\Delta T_{i,j,k} + \Delta T_{i,j,k+1})}{2}} \leq 0 \quad i \in HP, j \in CP, k \in ST \tag{24}$$

$$q_{CUi} - A_{CUi}U_{i,CU} \times \sqrt[3]{\Delta T_{CUi}(T^{H}_{OUTi} - T_{IN,CU})\frac{[\Delta T_{CUi} + (T^{H}_{OUTi} - T_{IN,CU})]}{2}} \leq 0 \quad i \in HP, j \in CP, k \in ST \tag{25}$$

$$q_{HUj} - A_{HUj}U_{j,HU} \times \sqrt[3]{\Delta T_{HUj}(T_{IN,HU} - T^{C}_{OUTj})\frac{[\Delta T_{HUj} + (T_{IN,HU} - T^{C}_{OUTj})]}{2}} \leq 0 \quad i \in HP, j \in CP, k \in ST \tag{26}$$

Additional variable bounds:

$$T^{H}_{OUTi} \leq T^{H}_{i,k} \leq T^{H}_{INi} \quad i \in HP \tag{27}$$

$$T^{C}_{OUTj} \geq T^{C}_{j,k} \geq T^{C}_{INj} \quad j \in CP \tag{28}$$

$$0 \leq q_{i,j,k} \leq \min\{CF^{H}_{i}(T^{H}_{INi} - T^{H}_{OUTi}), CF^{C}_{j}(T^{C}_{OUTj} - T^{C}_{INj})\} \quad i \in HP, j \in CP, k \in ST \tag{29}$$

$$0 \leq q_{CUi} \leq CF^{H}_{i}(T^{H}_{INi} - T^{H}_{OUTi}) \quad i \in HP \tag{30}$$

$$0 \leq q_{HUj} \leq CF^{C}_{j}(T^{C}_{OUTj} - T^{C}_{INj}) \quad j \in CP \tag{31}$$

4. LOWER BOUND MODEL

Because of the assumption of isothermal mixing, the model does not contain bilinear terms except when the area is calculated. We concentrate on the linearization of Chen's approximation of logarithmic mean temperature in eqs 24 to 26. We discretize the temperature differences $\Delta T_{i,j,k}$ as follows:

$$\sum_{\forall l} r_{i,j,k,l}\Delta T^{D}_{i,j,k,l} \leq \Delta T_{i,j,k} \leq \sum_{\forall l} r_{i,j,k,l}\Delta T^{D}_{i,j,k,l+1} \quad i \in HP, j \in CP, k \in ST \tag{32}$$

$$\sum_{\forall l} r_{i,j,k,l} = 1 \quad i \in HP, j \in CP, k \in ST \tag{33}$$

where the index l corresponds to the partitions. Then, we replace eq 24 with the following set:

$$q_{i,j,k} - \sum_{\forall l,\forall m} Z_{i,j,k,l,m}K_{i,j,k,l,m} \leq 0 \quad i \in HP, j \in CP, k \in ST \tag{34}$$

DOI: 10.1021/ie5032315
Ind. Eng. Chem. Res. XXXX, XXX, XXX−XXX

$$\sum_{\forall l} Z_{i,j,k,l,m} - \Psi r_{i,j,k,m} \le 0$$
$$i \in \text{HP}, j \in \text{CP}, k \in \text{ST}, m \in \text{L} \quad (35)$$

$$\sum_{\forall m} Z_{i,j,k,l,m} - \Psi r_{i,j,k+1,l} \le 0$$
$$i \in \text{HP}, j \in \text{CP}, k \in \text{ST}, l \in \text{L} \quad (36)$$

$$\sum_{\forall m, \forall l} Z_{i,j,k,l,m} - A_{i,j,k} = 0 \quad i \in \text{HP}, j \in \text{CP}, k \in \text{ST} \quad (37)$$

where

$$K_{i,j,k,l,m} = U_{i,j} \sqrt[3]{\Delta T^{D}_{i,j,k,l} \Delta T^{D}_{i,j,k+1,m} \frac{(\Delta T^{D}_{i,j,k,l} + \Delta T^{D}_{i,j,k+1,m})}{2}} \quad (38)$$

and $Z_{i,j,k,l,m}$ is binary. A similar relaxation can be written for equations 25 and 26.

The resulting model after the substitutions constitutes a mixed integer linear relaxed model that is a lower bound of the full model eq 1 through eq 31. An alternative would have been a reformulation of eqs 24−26 to obtain a quadratic set of equations (see Manousiouthakis and Sourlas[73]).

5. BOUND CONTRACTION

The bound contraction procedure used is the interval elimination strategy presented by Faria and Bagajewicz.[1] The basic strategy is summarized next. Further details of different strategies can be found in the original paper.

1. Run the lower bounding model (presented in section 4) to obtain a lower bound of the problem and identify the intervals containing the solution of the lower bounding model. Update the overall lower bound (OLB).
2. Run the original MINLP initialized by the solution of the lower bounding model (previous step) to find an upper bound solution. If a solution is found, update the overall upper bound value (OUB).
3. Calculate the gap between update OUB and OLB. If the gap is lower than the tolerance, the solution was found. Otherwise go to the step 4.
4. Run the lower bounding model forbidding the intervals selected in step 1. All previously forbidden intervals are set free. If the LB is infeasible, or its value is larger than the current OUB, then all the intervals that have not been forbidden for this variable are eliminated. The surviving feasible region between the new bounds is repartitioned.
5. Repeat step 4 for all the other variables, one at a time.
6. Go back to step 1 (a new iteration using contracted bounds starts).

Remarks:

- Different options for bound contracting have been proposed (Faria and Bagajewicz[1]): One-pass interval elimination, cyclic elimination, single and extended interval forbidding, etc., all of which are detailed in the referenced articles.
- The process is repeated with new bounds until convergence or until the bounds cannot be contracted anymore.
- If the bound contraction is exhausted, there are two possibilities to guarantee global optimality: Increase the partitioning of the variables to a level in which the sizes of the intervals are small enough to generate a lower bound within a given acceptable tolerance to the upper bound, or
- Recursively split the problem in two or more subproblems using a strategy such as the ones based on branch and bound procedure.

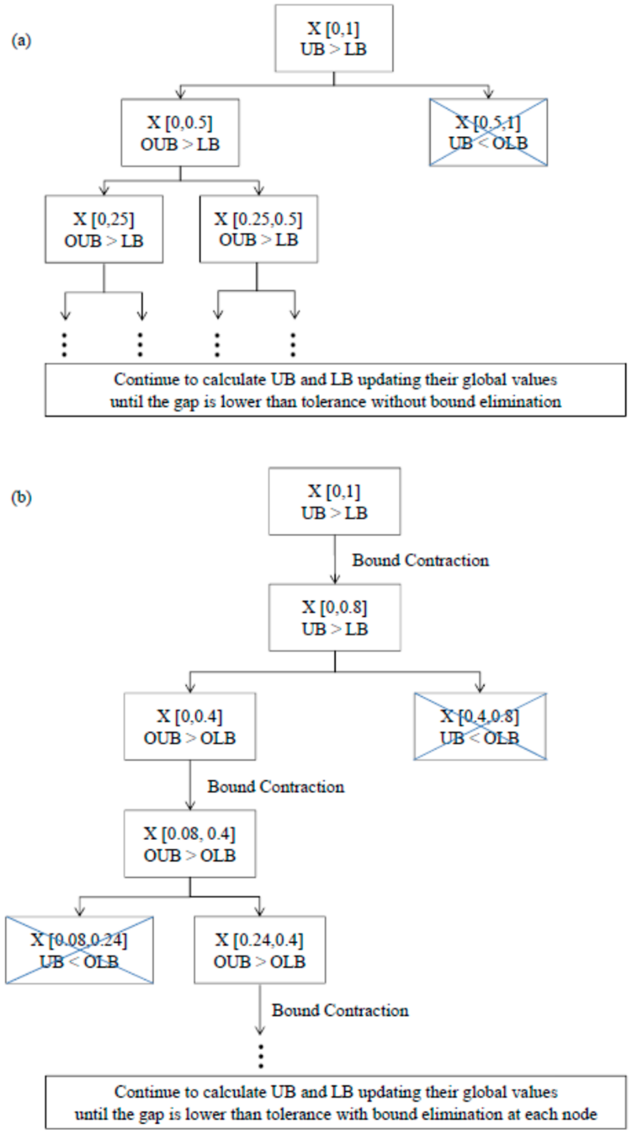

Figure 2. Illustration of the branch and bound procedures (a) without bound contraction and (b) with bound contraction at each node.

Table 1. Example 1 Data[a]

	T_S	T_T	h	CF	C
stream	K	K	kW/(m²K)	kW/K	$/(kWa)
H1	590	370	1.3	5	
C1	395	670	0.5	15	
CU	290	300	1.0		60
HU	680	680	5.0		120

[a]EMAT = 5 K.

Branch and Bound with Bound Contraction. The branch and bound procedures used with and without bound contraction by Faria and Bagajewicz[1] at each node are illustrated in Figure 2

DOI: 10.1021/ie5032315
Ind. Eng. Chem. Res. XXXX, XXX, XXX−XXX

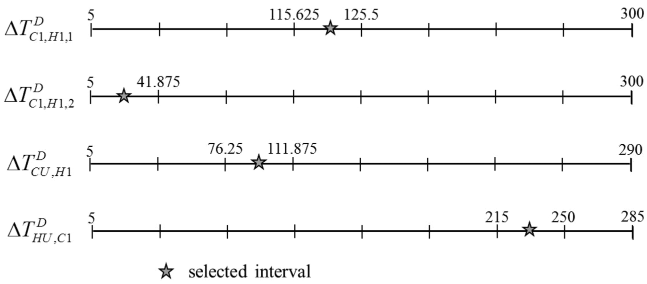

Figure 3. Selection of intervals in lower bounding MILP.

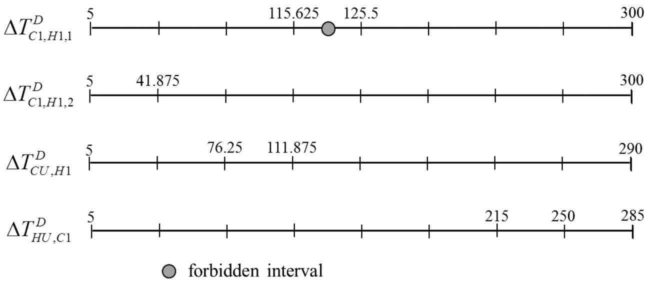

Figure 4. Interval elimination for $\Delta T^{\mathrm{D}}_{\mathrm{C1,H1,1}}$.

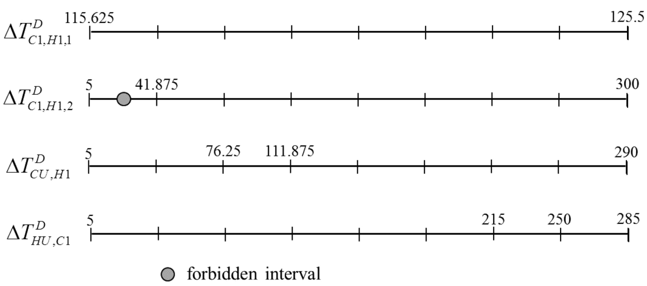

Figure 5. Interval elimination for $\Delta T^{\mathrm{D}}_{\mathrm{C1,H1,2}}$.

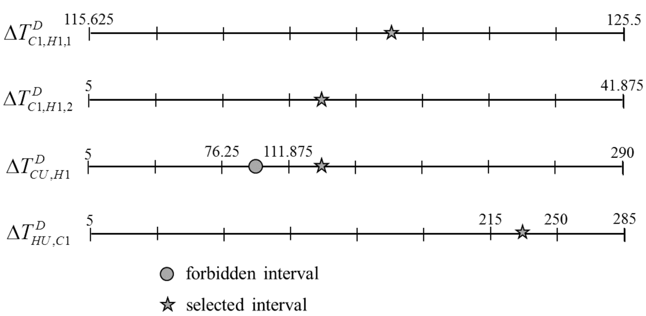

Figure 6. Interval elimination for $\Delta T^{\mathrm{D}}_{\mathrm{CU,H1}}$.

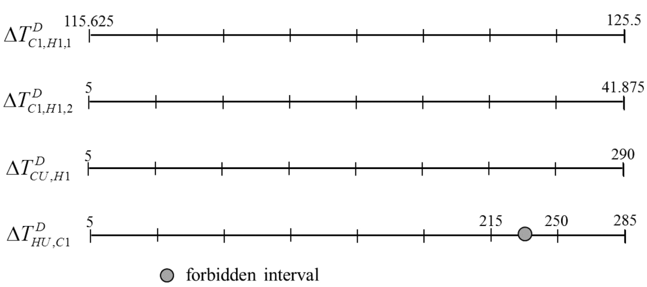

Figure 7. Interval elimination for $\Delta T^{\mathrm{D}}_{\mathrm{HU,C1}}$.

for one generic variable between zero and one. Both techniques are summarized next:

Table 2. Example 2 Data[a]

stream	T_S K	T_T K	h kW/(m²K)	CF kW/K	C \$/(kWa)
H1	650	370	1.0	10	
H2	590	370	1.0	20	
C1	410	650	1.0	15	
C2	350	500	1.0	13	
CU	300	320	1.0		15
HU	680	680	5.0		80

[a]EMAT = 10 K.

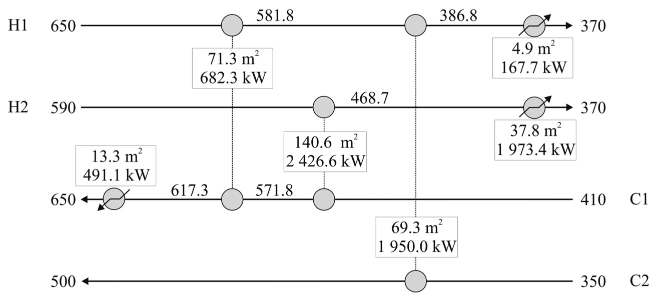

Figure 8. Example 2 solution.

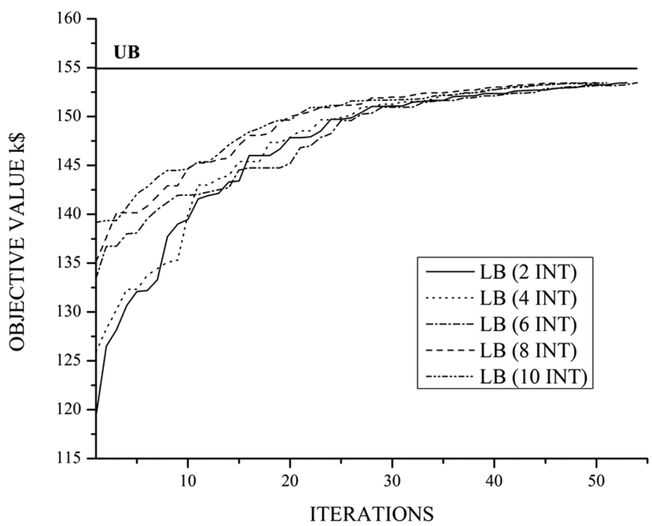

Figure 9. Progression of lower bound (Example 2).

Table 3. Example 3 Data[a]

stream	T_S K	T_T K	h kW/(m²K)	CF kW/K	C \$/(kWa)
H1	433.15	366.15	0.06	2.634	
H2	522.15	411.15	0.06	3.162	
H3	500.15	339.15	0.06	4.431	
H4	472.15	339.15	0.06	5.319	
C1	333.15	433.15	0.06	2.286	
C2	389.15	495.15	0.06	1.824	
C3	311.15	494.15	0.06	2.532	
C4	355.15	450.15	0.06	5.184	
C5	366.15	478.15	0.06	4.170	
CU	311.15	355.15	0.06		53349
HU	544.15	422.15	0.06		566167

[a]EMAT = 10 K.

1. Set the overall upper bound as OUB = +∞

DOI: 10.1021/ie5032315
Ind. Eng. Chem. Res. XXXX, XXX, XXX−XXX

Table 4. Example 3 Economic Data

	C_{HE} (\$)	C_A (\$/m^2)	C (\$/kWa)
HE	5291.9	77.79	
CU		77.79	53349
HU		77.79	566167

2. Solve the lower bound model and set the overall lower bound as the OLB as the current solution, if it is feasible. Otherwise, the original problem is infeasible.
3. Solve the upper bound model (in our case the original MINLP model) using the solution of the lower bound model as the starting point. If a solution is found, update OUB to its lower value.
4. If the gap between OUB and OLB is lower than the tolerance, the solution was found. Otherwise, set the first B&B node as the solution obtained in step 2 and go to steps a or b.

(a) Branch and bound without bound contraction
   1. Choose an open node.
   2. Divide a variable into two intervals and create two subproblems (or branch and bound nodes) and find their lower bounding solution.
   3. If the subproblem is feasible, go to next step. Otherwise, fathom this node and go to step a1.
   4. Run the upper bounding model for the feasible subproblem(s). If a new upper bound solution is found that is lower than the current OUB, update the OUB.
   5. Update the OLB as the minimum value of the open nodes lower bounds.
   6. Repeat last five steps until the difference between the OUB and the OLB is lower than the tolerance.

(b) Branch and bound with bound contraction at each node
   1. Choose an open node.
   2. Divide a variable into two subproblem (or new nodes) and find their lower bounding solution.
   3. Bound contract each subproblem.
   4. If the bound contracted subproblem becomes infeasible, this subproblem is infeasible and this node can be fathomed. Go to step b1.
   5. If the subproblem is feasible, run the upper bounding model for the feasible subproblem(s). If a new solution is found, update the OUB.
   6. Update the OLB as the minimum value of the open nodes lower bounds.
   7. Repeat last six steps until the difference between the OUB and the OLB is lower than the tolerance.

In the branch and bound with the bound contraction, our bound contraction procedure is applied to only one iteration at each node before variables are branched (Figure 2b). This method may sometimes take a longer or shorter time depending on the success of the bound contraction step.

Examples. Three examples of different sizes are presented in this section. The examples were implemented in GAMS (version 23.7)[74] and solved using CPLEX (version 12.3) as the MIP solver and DICOPT[75] as the MINLP solver on a PC machine (i7 3.6 GHz, 8 GB RAM).

Example 1. The first example is a small two-stream example used to illustrate the proposed approach in detail. For this reason, a relatively large number of partitioning intervals (eight) was chosen to ensure that only one interval elimination loop is needed to satisfy the convergence criterion in a few iterations.

The data for the example is presented in Table 1. The fixed cost of units is 10,000 \$, and the area cost coefficient is 350 \$/m^2. The number of continuous variables is 18, the number of binary variables is 4, the number on the number of equations is 23, the number of nonlinearly participating variables is 14, and the

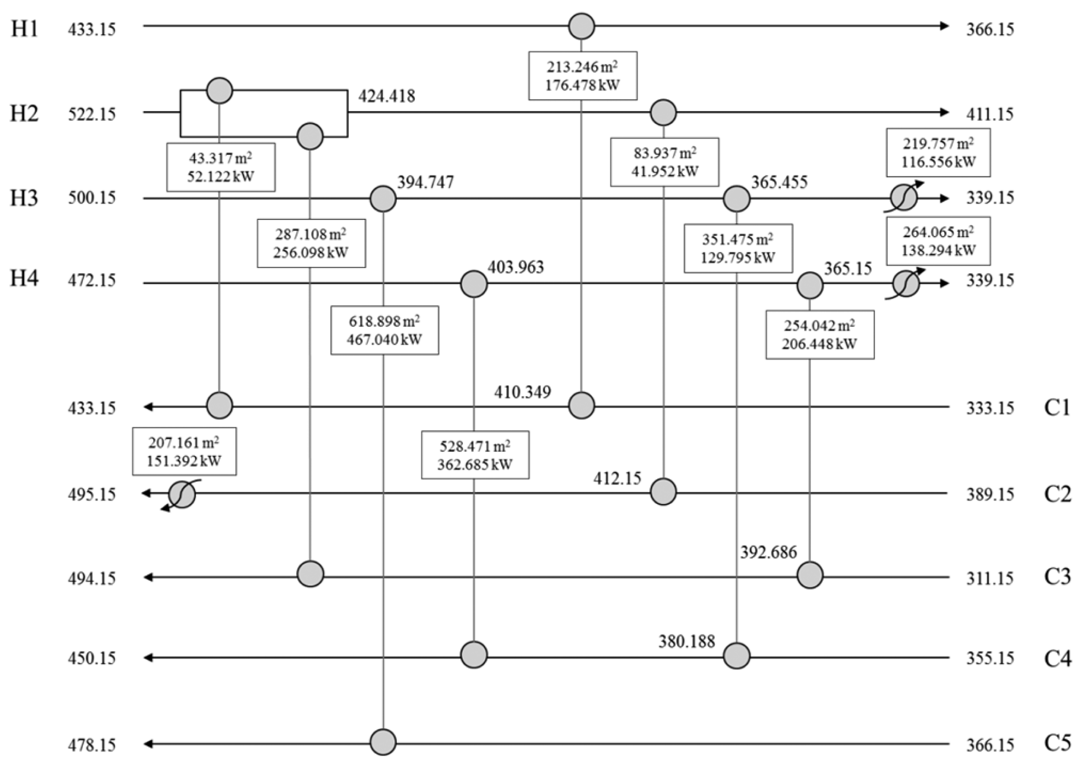

Figure 10. Example 3(10SP1) solution.

DOI: 10.1021/ie5032315
Ind. Eng. Chem. Res. XXXX, XXX, XXX−XXX

Table 5. Comparison of Results for Different Numbers of Partitioning Intervals

no. of intervals	2	3	4	5	6	7
objective value (k\$/year)	99606.28	99606.28	99606.28	99606.28	99615.19	99629.62
gap between UB and LB	0.099%	0.091%	0.085%	0.096%	0.072%	0.096%
no. of iterations	5	3	2	1	2	1
time	20 s	14 s	16 s	4 s	14 s	5 s

Table 6. Comparison of the Results with Different Methods

method	objective value(k $/year)	no. of iterations	GAP between UB and LB	time
bound contraction (with 2 intervals)	99606.28	5	0.099%	20 s
B&B without bound contraction at each node (with 2 intervals)	99606.28	100	0.144%	1 m 48 s
B&B with bound contraction at each node (with 2 intervals)	99606.28	6	0.099%	35 s

number of distinct nonlinear terms is 4. We used 1% gap between UB and LB as termination criteria and 8 partitioning intervals in the discretizing variable. The lower bounding MILP has 35 binary variables, of which 32 correspond to binary variables used for the interval partitioning, and 127 are continuous variables.

According to the solution algorithm described in the previous section, the lower bounding MILP is solved first to provide a lower bound, the first set of partitioning intervals, and initial point for the upper-bounding MINLP. The lower bound obtained is 467.330 k\$.

In Figure 3, initial intervals for the partitioned variables (temperature differences) are depicted, and the intervals selected by the lower bound model are denoted by a star.

Since this example consists of only two process streams and two utilities, only one heat exchanger, one cooler, and one heater are possible in a one-stage superstructure. Thus, only four temperature differences are variables which are subjected to partitioning. These are the temperature difference on a hot side ($k = 1$) of a heat exchanger ($\Delta T_{\mathrm{C1,H1,1}}$), the temperature difference on a cold side ($k = 2$) of a heat exchanger ($\Delta T_{\mathrm{C1,H1,2}}$), the temperature difference at the hot side of a cooler ($\Delta T_{\mathrm{CU,H1}}$), and the temperature difference at the cold side of a heater ($\Delta T_{\mathrm{HU,C1}}$). In this example, the tolerance gap between the lower and upper bound is 1%.

Next, the upper-bounding nonconvex MINLP is solved, giving an upper bound of 483.808 k\$. The gap between the lower and upper bound equals 3.406%. Because the convergence criterion (difference between OLB and OLB lower than the tolerance) is not satisfied, the elimination of intervals is performed in order to tighten the feasible region of a lower bounding problem.

In Figures 4−7 the *one-pass* interval elimination procedure is presented. The procedure eliminates intervals—one partitioned variable at a time, starting with $\Delta T^{\mathrm{D}}_{\mathrm{C1,H1,1}}$. First, the procedure states that the interval that was selected by the lower bounding MILP (see Figure 3) should be forbidden. Solving the lower bounding MILP produces the infeasibility of the lower bound model; all the intervals, except the one that was forbidden, can be eliminated. In other words, the feasible space for variable $\Delta T_{\mathrm{C1,H1,1}}$ is reduced from initial $\Delta T_{\mathrm{C1,H1,1}} \in [\mathrm{EMAT},\Gamma]$ to $\Delta T_{\mathrm{C1,H1,1}} \in [115.625,152.5]$. The surviving interval is repartitioned to the same number of intervals (eight).

The same procedure is repeated for the second partitioned variable $\Delta T_{\mathrm{C1,H1,2}}$, respectively. The available intervals are depicted in Figure 5. Note that the available discrete values for the temperature difference ($\Delta T_{\mathrm{C1,H1,1}}$) are the ones obtained after repartitioning in the previous elimination step (Figure 4).

The lower bound obtained is the infeasible result. All the allowed intervals ($\Delta T^{\mathrm{D}}_{\mathrm{C1,H1,2}} \geq 41.875$) are eliminated. The remaining interval is repartitioned.

Figure 6 shows the available intervals for the third partitioned variable ($\Delta T_{\mathrm{CU,H1}}$). Unlike in the first two interval elimination steps, a feasible solution is obtained with the objective value lower than the current upper bound.

Since the lower bound obtained in this elimination step (483.326 k\$) is not higher than the current upper bound (483.808 k\$), none of the intervals can be eliminated.

Finally, the last partitioned variable ($\Delta T^{\mathrm{D}}_{\mathrm{HU,C1}}$) is subjected to the elimination procedure. Figure 7 shows the available intervals.

Again, the lower bounding model with the forbidden interval produced the infeasible solution. The nonforbidden intervals are eliminated and the forbidden one is repartitioned. This concludes the first iteration.

In the second iteration, first, the lower bounding MILP is solved, yielding a lower bound of 479.564 k\$. Next, the upper-bounding nonconvex MINLP is solved. The upper bound obtained equals 483.808 k\$. The gap between lower and upper bound reduces from 3.406% (first iteration) to 0.877%. Since the convergence criterion is met the algorithm stops without triggering the second interval elimination loop.

The total CPU time needed to obtain the solution was 0.28 s, of which 33% corresponds to MINLP resource usage. Solving the same example, but fixing the integers and using NLP to obtain the upper bound, reduces the total time needed to 0.25 s.

Example 2. The second example consists of two hot and two cold streams. The data are given in Table 2. The fixed cost of units is 5500 \$, and the area cost coefficient is 150 \$/m^2. The number of continuous variables is 69, the number of binary variables is 12, the number on the number of equations is 81, the number of nonlinearly participating variables is 44, and the number of distinct nonlinear terms is 12. The lower bounding MILP has 44 binary variables, of which 28 correspond to binary variables used for the interval partitioning, and 125 are continuous variables. We used a 1% gap between UB and LB as termination criteria and 2 partitioning intervals in the discretizing variable.

The globally optimal solution, depicted in Figure 8, has an annualized cost of 154.995 k\$. It was obtained in 4 min 10 s of CPU time when we used 2 partitioning intervals.

The example was solved using a two-stage superstructure and different numbers of partitioning intervals (2, 4, 6, 8, and 10). Figure 9 shows the progression of lower bound values as a function of the iterations (one iteration is one cycle of bound contraction following the running of the unrestricted LB model and the UB model).

Example 3. The third example consisting of five cold and four hot process streams and one cooling and one heating utility was reported as 10SP1 in Cerda[76] as well as in Papoulias and Grossmann.[77] The data are given in Tables 3 and 4. The fixed

DOI: 10.1021/ie5032315
Ind. Eng. Chem. Res. XXXX, XXX, XXX−XXX

cost of units is 5291.9 \$, and the area cost coefficient is 77.79 $\$/m^2$. This example was solved using a two-stage superstructure. We used 0.1% gap between UB and LB as termination criteria and 2 partitioning intervals in the discretizing variable. The number of continuous variables is 264, the number of binary variables is 49, the number on the number of equations is 260, the number of nonlinearly participating variables is 187, and the number of distinct nonlinear terms is 49. The lower bounding MILP has 187 binary variables, of which 118 correspond to binary variables used for the interval partitioning, and 511 are continuous variables. DICOPT cannot find any optimum solution unless a good initial value is given.

The globally optimal solution, depicted in Figure 10, has an annualized cost of 99606.288 \$ and is obtained in the root node of the fifth iteration after 20 s of CPU time. The statistics for the different numbers of intervals used is shown in Table 5. As we add more intervals, the number of iterations tends to diminish and the time increases. Noticeably, this problem takes less time than the smaller example 2. This can be possibly explained by the existence of several suboptimal solutions close to the optimum for example 2 and not so many for the present case. We leave this analysis for future work.

We also tested the method, without using branch and bound and using branch and bound with and without bound contraction procedures. The results of these methods are summarized in Table 6. Note that the B&B methods were stopped at 100 iterations. Quite clearly, the additional number of MILP runs, does not make B&B a good option, even without the bound contraction at each node.

We also run the three examples using Antigone and Baron. Antigone solved problems 1, 2, and 3 to zero gap in 0.31 s, 1.5 s, and 4 min 54 s, respectively. In turn, Baron solved these problems in 0.03 s, 14 min 30 s, and 13 s, respectively. This was obtained using personal communications.[78,79] Comparisons are made on different machines, and, in addition, using algorithms (Antigone and Baron) that have years of development and improvement, while ours is still under development. Nevertheless, they compare well with our times: 0.28 s, 4 min 10 s, and 4 s for our strategy.

■ CONCLUSIONS

We presented a new bound contraction method based on domain and image partitioning for the global solution of MINLP problems. The key to the method is the lower bound model, which exhibits a tight bound. This model is then combined with pure bound contraction (as in Faria and Bagajewicz[1]), branch and bound (classical version) and branch and bound with bound contraction at each node. The model is significantly faster for the big problem that we tested as compared to the branch and bound options. In fact, we can say that for the problem tested, the branch and bound with or without bound contraction, using our lower bound model is not efficient.

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: bagajewicz@ou.edu. Tel: +1-405-325-5458.

Notes

The authors declare no competing financial interest.

■ NOMENCLATURE

Sets:

L, M, N, O = discretization grid
HP = hot process streams
CP = cold process streams
ST = stages
I = hot process streams
J = cold process streams
K = stages

Parameters:

$\Delta T^{D}_{i,j,k,l}$ = discretized temperature difference in stage k; match (i, j)
$\Delta T^{D}_{CU\ i,n}$ = discretized temperature difference on a hot side of a cooler; match (i, CU)
$\Delta T^{D}_{HU\ j,o}$ = discretized temperature difference on a cold side of a heater; match (j, CU)
M: = large value for "big M" constraints
Ψ = upper bound on areas
T_{IN} = inlet temperature
T_{OUT} = outlet temperature
C_{CU} = cold utilities cost
C_{HU} = cold utilities cost
C_{A} = area cost parameter
C_{HE} = fixed heat exchanger cost
CF = heat capacity flow rate
NOK = number of stages
Ω = upper bound on exchanged heat
Γ = upper bound on temperature difference
EMAT = exchanger minimum approach temperature

Variables:

$\Delta T_{i,j,k}$ = temperature difference in stage k for match between stream i and j
$\Delta T_{CU\ i}$ = temperature difference for match between stream i and cold utility
$\Delta T_{HU\ j}$ = temperature difference for match between stream j and hot utility
$q_{i,j,k}$ = exchanged heat for (i, j) match in stage k
$q_{CU\ i}$ = cold utility demand for stream i
$q_{HU\ j}$ = hot utility demand for stream j
$T^{H}_{i,k}$ = temperature of hot stream i on the hot side of stage k
$T^{C}_{j,k}$ = temperature of cold stream j on the hot side of stage k
$z_{i,j,k}$ = binary variable for (i, j) match in stage k
z_{cui} = binary variable for cold utility in stream i
z_{huj} = binary variable for hot utility in stream j

■ REFERENCES

(1) Faria, D.; Bagajewicz, M. A new approach for global optimization of a class of MINLP problems with applications to water management and pooling problems. *AIChE J.* **2012**, *58* (8), 2320−2335.
(2) Yee, T. F.; Grossmann, I. E. Simultaneous optimization models for heat integration II. *Comput. Chem. Eng.* **1990**, *14*, 1165−1184.
(3) Sahinidis, N. V. BARON: A general purpose global optimization software package. *J. Global Optim.* **1996**, 7 (4), 337−363.
(4) Schichl, H. *COCOS—Mathematical modeling and global optimization*. Habilitation thesis, Cambridge University Press, 2003.
(5) Kearfott, R. B. Globsol user guide. *J. Optim. Methods Software, Global Optim.* **2009**, *24* (4−5), 687−708.
(6) Lebbah, Y. ICOS: a branch and bound based solver for rigorous global optimization. *J. Optim. Methods Software, Global Optim.* **2009**, *24* (4−5), 709−726.
(7) Pinter, J. D. Nonlinear optimization with GAMS/LGO. *J. Global Optim.* **2007**, *38* (1), 79−101.
(8) Schrage, L. *LINGO—Optimization Modeling with LINGO*, 6th ed.; LINDO Systems Inc.: Chicago, IL, 2006.
(9) Ugray, Z.; Lasdon, L.; Plummer, J.; Glover, F.; Kelly, J.; Marti, R. OQNLP—A multistart scatter search heuristic for smooth NLP and

DOI: 10.1021/ie5032315
Ind. Eng. Chem. Res. XXXX, XXX, XXX−XXX

MINLP problems. *Oper. Res./Comput. Sci. Interfaces Ser.* **2005**, *30*, 25−57.

(10) Baker, K. R.; Camm, J. D. Premium Solver—On the use of integer programming versus evolutionary solver in spreadsheet optimization. *Inf. Trans. Educ.* **2005**, *5* (3), 1−7.

(11) Androulakis, P.; Maranas, C. D.; Floudas, C. A. αBB: A global optimization method for general constrained nonconvex problems. *J. Global Optim.* **1995**, *7* (4), 337−363.

(12) Misener, R.; Floudas, C. A. GloMIQO: Global mixed-integer quadratic optimizer. *J. Global Optim.* **2013**, *57*, 3−50.

(13) Misener, R.; Floudas, C. A. ANTIGONE: Algorithms for continuous/integer global optimization of nonlinear equations. *J. Global Optim.* **2014**, In Press (DOI: 10.1007/s10898-014-0166-2).

(14) Achterberg, T.; Berthold, T.; Koch, T.; Wolter, K. Constraint Integer Programming: A New Approach to Integrate CP and MIP. Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. *Lect. Notes Comput. Sci.* **2008**, *5015*, 6−20.

(15) Achterberg, T. SCIP: solving constraint integer programs. *Math. Program. Comput.* **2009**, *1* (1), 1−41.

(16) Belloti, P.; Lee, J.; Liberti, L.; Margot, F.; Wächter, A. Branching and bounds tightening techniques for non-convex MINLP. *Optim. Methods Software* **2009**, *24* (4−5), 597−634.

(17) Sherali, H. D.; Adams, W. P. A reformulation-linearization technique for solving discrete and continuous nonconvex problems. *Nonconvex Optimization and Its Applications*; Kluwer Academic Publishers: Boston, MA, 1999.

(18) Floudas, C. A. Deterministic global optimization: theory, methods and applications. *Nonconvex Optimization and Its Applications*. Kluwer Academic Publishers: Boston, MA, 2000.

(19) Tawarmalani, M.; Sahinidis, N. V. Convexification and global optimization in continuous and mixed-integer nonlinear programming: theory, algorithms, software, and applications. *Nonconvex Optimization and Its Applications*. Kluwer Academic Publishers: Boston, MA, 2002.

(20) Horst, R.; Tuy, H. *Global Optimization: Deterministic Approaches*; Springer: New York, 2003.

(21) Hansen, E.; Walster, G. W. Global optimization using interval analysis. *Pure and Applied Mathematics*; Marcel Dekker: New York, 2004.

(22) Floudas, C. A. Global optimization in design and control of chemical process systems. *J. Process Control* **2000**, *10*, 125.

(23) Pardalos, P. M.; Romeijn, H. E.; Tuy, H. Recent developments and trends in global optimization. *J. Comput. Appl. Math.* **2000**, *124* (1−2), 209−228.

(24) Floudas, C. A.; Akrotirianakis, I. G.; Caratzoulas, S.; Meyer, C. A.; Kallrath, J. Global optimization in the 21st century: Advances and challenges. *Comput. Chem. Eng.* **2005**, *29* (6), 1185−1202.

(25) Floudas, C. A.; Gounaris, C. E. A review of recent advances in global optimization. *J. Global Optim.* **2009**, *45*, 3−38.

(26) Ben-Tal, A.; Eiger, G.; Gershovitz, V. Global minimization by reducing the duality gap. *Math. Program.* **1994**, *63*, 193−212.

(27) Adhya, N.; Tawarmalani, M.; Sahinidis, N. V. A Lagrangian approach to the pooling problem. *Ind. Eng. Chem. Res.* **1999**, *38*, 1956−1972.

(28) Kuno, T.; Utsunomiya, T. A Lagrangian based branch-and-bound algorithm for production-transportation problems. *J. Global Optim.* **2000**, *18*, 59−73.

(29) Karrupiah, R.; Grossmann, I. E. A Lagrangian based branch-and-cut algorithm for global optimization of nonconvex mixed-integer nonlinear programs with decomposable structures. *J. Global Optim.* **2008**, *41*, 1573−2916.

(30) Ruiz, J. P.; Grossmann, I. E. Strengthening of lower bounds in the global optimization of bilinear and concave generalized disjunctive programs. *Comput. Chem. Eng.* **2010**, *34* (6), 914−930.

(31) Westerlund, T.; Skifvars, H.; Harjunkoski, I.; Pörn, R. An extended cutting plane method for a class of nonconvex MINLP problems. *Comput. Chem. Eng.* **1998**, *22* (3), 357−365.

(32) Moore, R. E. *Interval Analysis*; Prentice-Hall: 1966.

(33) Hansen, E. R. Global optimization using interval analysis: The one-dimensional case. *J. Optim. Theory Appl.* **1979**, *29*, 331−344.

(34) Ratschek, H.; Rokne, J. *New Computer Methods for Global Optimization*; Halsted Press: New York, NY, 1988.

(35) Moore, R.; Hansennd, E. R.; Leclerc, A. *Rigorous Methods for Global Optimization. Recent Advances in Global Optimizarion;* Floudas, C. A.; Pardalos, P. M., Eds.; Princeton University Press, Princeton, NJ, 1992; p 321.

(36) Vaidyanathan, R.; El-Halwagi, M. Global optimisation of nonconvex programs via interval analysis. *Comput. Chem. Eng.* **1994**, *18*, 889−897.

(37) Byrne, R. P.; Bogle, I. D. L. Global optimization of constrained nonconvex programs using reformulation and interval analysis. *Comput. Chem. Eng.* **1999**, *23*, 1341−1350.

(38) Ryoo, H. S.; Sahinidis, N. V. Global optimization of nonconvex NLPs and MINLPs with applications in process design. *Comput. Chem. Eng.* **1995**, *19* (5), 551−566.

(39) Adjiman, C. S.; Androulakis, I. P.; Floudas, C. A. Global optimization of MINLP problems in process synthesis and design. *Comput. Chem. Eng.* **1997**, *21*, S445−S450.

(40) Smith, E. M. B.; Pantelides, C. C. Global optimization of nonconvex MINLPs. *Comput. Chem. Eng.* **1997**, *21*, 791−S796.

(41) Kesavan, P.; Barton, P. I. Generalized branch-and-cut framework for mixed-integer nonlinear optimization problems. *Comput. Chem. Eng.* **2000**, *4* (2/7), 1361−1366.

(42) Zamora, J. M.; Grossmann, I. E. A global MINLP optimization algorithm for the synthesis of heat exchanger networks with no stream splits. *Comput. Chem. Eng.* **1998**, *22*, 367−384.

(43) Zamora, J. M.; Grossmann, I. E. A branch and contract algorithm for problems with concave univariate, bilinear and linear fractional terms. *J. Global Optim.* **1999**, *14*, 217−249.

(44) Geoffrion, A. M. Generalized Benders decomposition. *J. Optim. Theory Appl.* **1972**, *10*, 237−262.

(45) Duran, M. A.; Grossmann, I. E. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Math. Program.* **1986**, *36*, 307−339.

(46) Fletcher, R.; Leyffer, S. Solving mixed integer nonlinear programs by outer approximation. *Mathematical Programming* **1994**, *66*, 327−349.

(47) Bagajewicz, M.; Manousiouthakis, V. On the generalized Benders decomposition. *Comput. Chem. Eng.* **1991**, *15* (10), 691−700.

(48) Sahinidis, N. V.; Grossmann, I. E. Convergence properties of generalized benders decomposition. *Comput. Chem. Eng.* **1991**, *15* (7), 481−491.

(49) Kesavan, P.; Barton, P. I. Decomposition algorithms for nonconvex mixed-integer nonlinear programs. *AIChE Symp. Ser.* **2000**, *96* (323), 458−461.

(50) Faria, D.; Bagajewicz, M. Global optimization of water management problems using linear relaxations and bound contraction methods. *Ind. Eng. Chem. Res.* **2011**, *50* (7), 3738−3753.

(51) Faria, D.; Bagajewicz, M. Novel bound contraction procedure for global optimization of bilinear MINLP problems with applications to water management problems. *Comput. Chem. Eng.* **2011**, *35*, 446−455.

(52) Faria, D.; Bagajewicz, M. Global optimization based on sub-spaces elimination. Applications to generalized pooling and water management problems. *AIChE J.* **2012**, *58* (8), 2336−2345.

(53) Sherali, H. D.; Alamedinne, A. A new reformulation-linearization technique for bilinear programming problems. *J. Glob. Optim.* **1992**, *2* (4), 379−410.

(54) Meyer, C. A.; Floudas, C. A. Global optimization of a combinatorially complex generalized pooling problem. *AIChE J.* **2006**, *52−3*, 1027−1037.

(55) Misener, R.; Floudas, C. A. Advances for the pooling problem: Modeling, global optimization, and computational studies. *Appl. Comput. Math.* **2009**, *8* (1), 3−22.

(56) Misener, R.; Floudas, C. A. Global optimization of large-scale generalized pooling problems: Quadratically constrained MINLP models. *Ind. Eng. Chem. Res.* **2010**, *49*, 5424−5438.

(57) Misener, R.; Thompson, J. P.; Floudas, C. A. Apogee: Global optimization of standard, generalized, and extended pooling problems

via linear and logarithmic partitioning schemes. *Comput. Chem. Eng.* **2011**, *35*, 876−892.
(58) Misener, R.; Floudas, C. A. Global optimization of mixed-integer quadratically-constrained quadratic programs (MIQCQP) through piecewise-linear and edge-concave relaxations. *Mathematical Programming* **2012**, *136*, 155−182.
(59) Misener, R.; Floudas, C. A. GloMIQO: Global mixed-integer quadratic optimizer. *J. Global Optim.* **2013**, *57*, 3−50.
(60) Gounaris, C. E.; Misener, R.; Floudas, C. A. Computational comparison of piecewise-linear relaxations for poolings. *Ind. Eng. Chem. Res.* **2009**, *48*, 5742−5766.
(61) Karuppiah, R.; Grossmann, I. E. Global optimization for the synthesis of integrated water systems in chemical processes. *Comput. Chem. Eng.* **2006**, *30*, 650−673.
(62) Karuppiah, R.; Grossmann, I. E. Global optimization of multiscenario mixed integer nonlinear programming models arising in the synthesis of integrated water networks under uncertainty. *Comput.-Aided Chem. Eng.* **2006**, *21* (2), 1747−1752.
(63) Bergamini, M. L.; Aguirre, P.; Grossmann, I. Logic based outer approximation for global optimization of synthesis of process networks. *Comput. Chem. Eng.* **2005**, *29*, 1914−1933.
(64) Bergamini, M. L.; Grossmann, I.; Scenna, N.; Aguirre, P. An improved piecewise outer-approximation algorithm for the global optimization of MINLP models involving concave and bilinear terms. *Comput. Chem. Eng.* **2008**, *32*, 477−493.
(65) Wicaksono, D. A.; Karimi, I. A. Piecewise MILP under- and overestimators for global optimization of bilinear programs. *AIChE J.* **2008**, *54* (4), 991−1008.
(66) Hasan, M. M. F.; Karimi, I. A. Piecewise linear relaxation of bilinear programs using bivariate partitioning. *AIChE J.* **2010**, *56* (7), 1880−1893.
(67) Pham, V.; Laird, C.; El-Halwagi, M. Convex hull discretization approach to the global optimization of pooling problems. *Ind. Eng. Chem. Res.* **2009**, *48* (4), 1973−1979.
(68) Zamora, J. M.; Grossmann, I. E. A comprehensive global optimization approach for the synthesis of heat exchanger networks with no stream splits. *Comput. Chem. Eng.* **1997**, *21*, 65−70.
(69) Adjiman, C. S.; Androulakis, I. P.; Floudas, C. A. Global optimization of mixed integer nonlinear problems. *AIChE J.* **2000**, *46*, 1769−1797.
(70) Björk, K.-M.; Westerlund, T. Global optimization of heat exchanger network synthesis problems with and without the isothermal mixing assumption. *Comput. Chem. Eng.* **2002**, *26*, 1581−1593.
(71) Bergamini, M. L.; Scenna, N. J.; Aguirre, P. A. Global optimal structures of heat exchanger networks by piecewise relaxation. *Ind. Eng. Chem. Res.* **2007**, *46*, 1752−1763.
(72) Barbaro, A.; Bagajewicz, M. J. New rigorous one-step MILP formulation for heat exchanger network synthesis. *Comput. Chem. Eng.* **2005**, *29*, 1945−1976.
(73) Manousiouthakis, V.; Sourlas, D. A Global optimization approach to rationally constrained rational programming. *Chem. Eng. Commun.* **1992**, *115*, 127−147.
(74) Brooke, A.; Kendrick, D.; Meeraus, D.; Raman, R. *GAMS—A User guide*; GAMS Development Corporation: Washington D.C., 2007.
(75) Viswanathan, J.; Grossmann, I. E. A combined penalty function and outer approximation method for MINLP optimization. *Comput. Chem. Eng.* **1990**, *14*, 769−782.
(76) Cerda, J. *Transportation models for the optimal synthesis of heat exchanger networks*. Ph.D. Thesis. Carnegie-Melon University: Pittsburgh, PA, 1980.
(77) Papoulias, S. A.; Grossmann, I. E. A structural optimization approach in process synthesis-II: Heat recovery networks. *Comput. Chem. Eng.* **1983**, *7* (6), 707−721.
(78) Floudas, C. *Personal communication*, 2014.
(79) Sahinidis, N. *Personal communication*, 2014.

DOI: 10.1021/ie5032315
Ind. Eng. Chem. Res. XXXX, XXX, XXX−XXX