Computers and Chemical Engineering 35 (2011) 446–455

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Novel bound contraction procedure for global optimization of bilinear MINLP problems with applications to water management problems

Débora C. Faria, Miguel J. Bagajewicz *

University of Oklahoma, 100 E. Boyd St., T335, Norman, OK 73019, USA

ARTICLE INFO

Article history:
Received 25 November 2009
Received in revised form 6 April 2010
Accepted 8 April 2010
Available online 14 April 2010

Keywords:
Mathematical programming
Optimization
Global optimization

ABSTRACT

We propose a new method to obtain the global optimum of MINLP problems containing bilinearities. Our special method that contracts the bounds of one variable at a time allows reducing the gap between a linear lower bound and an upper bound obtained solving the original problem. Unlike some methods based on variable partitioning, our bound contraction procedure does not introduce new integers or intervals. We illustrate the method by applying it to water management problems.

1. Introduction

Commercially available global optimization packages exist: BARON (Sahinidis, 1996), COCOS, GlobSol, ICOS, LGO, LINGO, OQNLP, Premium Solver. Other methods that have been applied to several engineering problems are known: αBB (Androulakis, Maranas, & Floudas, 1995), for example. Several books discuss and classify global optimization methods: Floudas (2000a), Hansen and Walster (2004), Horst and Tuy (2003), Sherali and Adams (1999), and Tawarmalani and Sahinidis (2002). There are also recent review papers: Floudas (2000b), Floudas, Akrotirianakis, Caratzoulas, Meyer, and Kallrath (2005), Floudas and Gounaris (2009), and Pardalos, Romeijn, and Tuy (2000).

There are a few different approaches for global optimization: one important class of methods are those based on branch and bound on key variables (Zamora & Grossmann, 1999) complemented by variable discretization (Bergamini, Aguirre, & Grossmann, 2005; Bergamini, Grossmann, Scenna, & Aguirre, 2008; Karuppiah & Grossmann, 2006a; Karuppiah & Grossmann, 2006b; Meyer & Floudas, 2006). Specifically, the papers by Bergamini et al. (2005, 2008) and Karuppiah and Grossmann (2006a, 2006b) focus on applications water problems, the applications we use for illustrations. Many other water problems that our group solved using mathematical programming are depicted in Faria and Bagajewicz (2006, 2008, 2009, 2010a, 2010b, 2010c) and Savelski and Bagajewicz (2001). Other approaches, some that also make use of some branch and bound techniques are: Lagrangean-based approaches (Adhya, Tawarmalani, & Sahinidis, 1999; Ben-Tal, Eiger, & Gershovitz, 1994; Karrupiah & Grossmann, 2008; Kuno & Utsunomiya, 2000), disjunctive programming-based methods (Ruiz & Grossmann, 2010) and intervals analysis arithmetic (Hansen, 1979; Moore, 1966; Moore, Hansen, & Leclerc, 1992; Ratschek & Rokne, 1988; Vaidyanathan & El-Halwagi, 1994).

Faria and Bagajewicz (2010c) reviewed in more detail some of the methods that involve variable discretization to construct lower bound MILP estimators. They also present a bound contraction procedure that, if sufficient discretization is used, can identify the global optimum without resorting to any branch and bound techniques.

One of the computational burdens in the method based on variable discretization (Faria & Bagajewicz, 2010c; Karuppiah & Grossmann, 2006a, 2006b; Meyer & Floudas, 2006) is that to become efficient, they may need a large number of intervals and this requires one binary variable for each. As a result large scale and time consuming MILP or MINLP problems have to be solved. The method we present in this paper is based on bound contraction without the need for such discretization and consequently does not need to use integers.

The paper is organized as follows: we present the relaxation methodology first, followed by the bound contraction procedure. We then present the global optimization algorithm followed by an extended bound contraction method. We finish with illustrations.

* Corresponding author at: Dept. of Chemical & Material Sciences, University of Oklahoma, 100 E. Boyd, T-335, Norman, OK 73019-0628, USA. Tel.: +1 405 325 5458; fax: +1 405 325 5813.

E-mail address: bagajewicz@ou.edu (M.J. Bagajewicz).

0098-1354/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compchemeng.2010.04.010

Nomenclature

Sets

u	water using units
w	freshwater sources
S	wastewater sinks
r	regeneration processes
C	contaminants

Parameters

fwc_w	freshwater cost
$\mathrm{RegOC_r}$	coefficient of the operating cost of regeneration processes
$\mathrm{RefCC_r}$	coefficient of the capital cost of regeneration processes
$F_{\max}$	maximum flowrate allowed through the connections
$F_{\min}$	minimum flowrate allowed through the connections
$CW_{w,c}$	freshwater concentrations
$C_{u,c}^{\text{in,max}}$	maximum inlet concentrations at each water-using unit
$C_{u,c}^{\text{out,max}}$	maximum outlet concentrations at each water-using unit
$CS_{s,c}^{\text{out,max}}$	maximum concentrations at the sinks
$\Delta M_{u,c}$	mass load of each contaminant at each water-using unit
$LWU_{w,u}$	distance between freshwater sources and water-using units
$LUU_{u,u*}$	distance between two water-using units
$LUR_{u,r}$	distance between water-using units and regeneration processes
$LUS_{u,s}$	distance between a water-using unit and a wastewater sink
$LWR_{w,r}$	distance between freshwater sources and regeneration processes
$LRR_{r,r*}$	distance between two regeneration processes
$LRU_{r,u}$	distance between regeneration processes and water-using units
$LRS_{r,s}$	distance between regeneration processes and wastewater sinks
β	fixed part of the connection capital cost (linear coefficient of the cost equation as function of flowrate)
α	variable part of the connection capital cost (angular coefficient of the cost equation as function of flowrate)
γ	power coefficient of the capital cost equation
df	discount factor

Variables

$FWU_{w,u}$	flowrate between freshwater source and water-using unit
$FUU_{u,u*}$	flowrate between two water-using units
$FUR_{u,r}$	flowrate between water-using unit and regeneration process
$FUS_{u,s}$	flowrate between water-using unit and wastewater sink
$FWR_{w,r}$	flowrate between freshwater source and regeneration process
$FRR_{r,r*}$	flowrate between two regeneration processes
$FRU_{r,u}$	flowrate between regeneration process and water-using unit
$FRS_{r,s}$	flowrate between regeneration process and wastewater sink
$Consu$	freshwater consumption
$CU_{u,c}$	outlet concentration of water-using units
$CR_{r,c}$	outlet concentration of regeneration processes

Binary variables

$YWU_{w,u}$	existence of connection between freshwater sources and water-using units
$YUU_{u,u*}$	existence of connection between two water-using units
$YUR_{u,r}$	existence of connection between water-using units and regeneration processes
$YUS_{u,s}$	existence of connection between a water-using unit and a wastewater sink
$YWR_{w,r}$	existence of connection between freshwater sources and regeneration processes
$YRR_{r,r*}$	existence of connection between two regeneration processes
$YRU_{r,u}$	existence of connection between regeneration processes and water-using units
$YRS_{r,s}$	existence of connection between regeneration processes and wastewater sinks

2. Relaxation methodology

Consider z to be the product of two continuous variables x and y:

$$z_{ij} = y_j x_i \quad \forall i = 1, \ldots, n; \quad \forall j = 1, \ldots, m \tag{1}$$

where both x_i and y_j are subject to certain bounds:

$$x_i^L \le x_i \le x_i^U \quad \forall i = 1, \ldots, n \tag{2}$$

$$y_j^L \le y_j \le y_j^U \quad \forall j = 1, \ldots, m \tag{3}$$

We propose to replace Eq. (1) by the following two equations:

$$z_{ij} \ge y_j \bar{x}_i^L \quad \forall i = 1, \ldots, n; \quad \forall j = 1, \ldots, m \tag{4}$$

$$z_{ij} \le y_j \bar{x}_i^U \quad \forall i = 1, \ldots, n; \quad \forall j = 1, \ldots, m \tag{5}$$

where we use updated bounds for x_i ($\bar{x}_i^L$ and $\bar{x}_i^U$). We notice that the bounds for x_i remain included in the problem. Likewise, while bounds for x_i are used in (4) and (5), this variable remains as such in any other linear term and is not substituted by any bounds. This particular choice of the lower bound model is not new: we use direct discretization, but other kind of relaxations, like concave and convex envelopes (McCormick, 1976) for example, could be used. Our novelty does not reside in this particular choice of relaxation, but on the use of reference values and the algorithm used to perform the bound contraction.

Because Eq. (1) is replaced by the relaxation Eqs. (4) and (5), the proposed problem is LP (or MILP if there are integer/binary variables in other linear terms) and is also a lower bound of the original problem. The method that we propose updates the bounds one variable at a time.

For reasons that will become clear later, we now introduce reference values that are calculated after a lower bound is obtained using the relaxed model. Let $\hat{z}_{ij}$, $\hat{y}_j$ and $\hat{x}_i$ be the results of the lower bound problem. Then, we define reference values for x_i $\left(x_i^{ref}\right)$, which are obtained as follows:

$$x_i^{ref} = f_x^{(w)}(\hat{z}_{i1}, \hat{z}_{i2}, \ldots, \hat{z}_{im}; \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m) \quad \forall i = 1, \ldots, n \tag{6}$$

We assume, for a moment that x_i^{ref} is a better estimate of x_i than $\hat{x}_i$. We now discuss the different forms of the function $f_x^{(w)}(\bullet)$. They

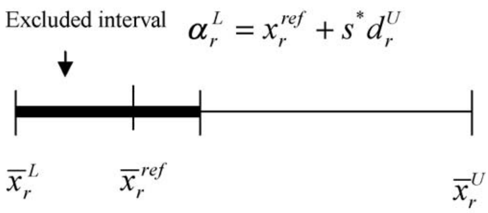

Fig. 1. Interval exclusion for bound contraction.

are

$$f_x^{(1)}(\hat{z}_{i1}, \hat{z}_{i2}, \ldots, \hat{z}_{im}; \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m) = \frac{\sum_{j=1,\ldots m} \hat{z}_{ij}}{\sum_{j=1,\ldots m} \hat{y}_j} \quad \forall i = 1, \ldots, n \tag{7}$$

$$f_x^{(2)}(\hat{z}_{i1}, \hat{z}_{i2}, \ldots, \hat{z}_{im}; \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m) = \underset{\forall j=1,\ldots m}{Max} \left\{ \frac{\hat{z}_{ij}}{\hat{y}_j} \right\} \quad \forall i = 1, \ldots, n \tag{8}$$

$$f_x^{(3)}(\hat{z}_{i1}, \hat{z}_{i2}, \ldots, \hat{z}_{im}; \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m) = \underset{\forall j=1,\ldots m}{Min} \left\{ \frac{\hat{z}_{ij}}{\hat{y}_j} \right\} \quad \forall i = 1, \ldots, n \tag{9}$$

$$f_x^{(4)}(\hat{z}_{i1}, \hat{z}_{i2}, \ldots, \hat{z}_{im}; \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m) = \underset{\forall j=1,\ldots m}{Average} \left\{ \frac{\hat{z}_{ij}}{\hat{y}_j} \right\} \quad \forall i = 1, \ldots, n \tag{10}$$

In the present work, we use $f_x^{(2)}$ and $f_x^{(3)}$ only. We also introduce the notion of lower and upper departure. These are distances to the bounds defined as follows:

$$d_i^L = x_i^{ref} - \bar{x}_i^L \quad \forall i = 1, \ldots, n \tag{11}$$

$$d_i^U = \bar{x}_i^U - x_i^{ref} \quad \forall i = 1, \ldots, n \tag{12}$$

We note that bilinear terms show up frequently in component balances. If x represents concentrations, and y are flowrates, while n is the number of components we can use Eq. (7) to find an average value of x and/or Eqs. (8) and (9) to define maximum and minimum values. In such case x_i^{ref} represents a new reference value for concentrations. In turn, if in these problems x represents flowrates, then x_i^{ref} represents a new reference value for the flowrates. In this case, Eq. (10) is used to define the average value instead.

3. Bound contraction procedure

We based our methodology on updating the bounds $\bar{x}_i^L$ and $\bar{x}_i^U$ for each variable one at a time. We assume first that the lowest departure is d_r^L, that is, x_r^{ref} is closer to $\bar{x}_r^L$ than to $\bar{x}_r^U$ and we define the Auxiliary Linear model ALB_r^L as one where the original bilinear constraint (1) for all variables is replaced by Eqs. (4) and (5), with the exception of Eq. (1) for z_{rj}, which is replaced by Eq. (5) as above and a modified Eq. (4) as follows:

$$z_{rj} \geq \alpha_r^L y_j \quad \forall j = 1, \ldots, m \tag{13}$$

In turn, α_r^L is given by

$$\alpha_r^L = x_r^{ref} + s d_r^U, \tag{14}$$

where s can vary from 0 to 0.99.

Thus, we run problem ALB_r^L for different incremental increasing values of s (Δs) until one reaches a point where the problem is infeasible or this lower bound is higher than the current upper bound for a certain $s = s^*$. This is illustrated in Fig. 1. We also update the lower bound for ALB_r^L to be $x_i^L = \alpha_r^L$.

Several different strategies can be implemented to determine s^*. One could start with $s = 0$ and keep increasing s until s^* is identified or s is equal a pre-defined $s^{\max}$. However, this strategy may take too many steps, especially when Δs is small. One alternative is to start with some value of s, say $s = \varepsilon$ and x_r^{ref} equal to $f_x^{(2)}$. The reason for this is that $f_x^{(2)}$ is the best estimate of the largest reference value for x_i and therefore the excluded interval may contain all the possible solutions for x_i. If one sets x_r^{ref} equal to $f_x^{(1)}$ or $f_x^{(3)}$, the relaxed terms may have values on the non-forbidden portion of x_i.

Quite clearly, there is a compromise between the size of Δs, or the chosen x_r^{ref} and the strategy to use. In the latter case the value of $x_r^{ref} = f_x^{(3)}$ may be too low and too many steps may be needed until an interval bound contraction is performed. However, if contraction happens earlier, the procedure improves quickly because is more efficient. In the former case, the chances of eliminations in earlier iterations is higher, but the improvement of the bound contraction is slower due to eliminations of smaller portions of the x_i. We opt for the simple case of starting with the suggested value of $x_r^{ref} = f_x^{(3)}$, starting with $s = \varepsilon$, and march forward if needed. Note that x_r^{ref} must never be smaller than $f_x^{(3)}$.

Thus, at this point one can say that with all the current bounds in place for all variables, one can be certain that the solution of the problem does not contain a value of x_i in the interval $[x_i^{ref} + s d_i^U, \bar{x}_r^U]$ and therefore that portion of the feasible space can be eliminated. In other words one should update the upper bound as follows: $\bar{x}_r^U \leftarrow x_i^{ref} + s d_i^U$.

When the lowest departure is d_r^U, that is, x_r^{ref} is closer to $\bar{x}_r^U$ than to $\bar{x}_r^L$, we define the Auxiliary Linear model ALB_r^U, where instead of modifying Eq. (5) for $i = r$, we modify Eq. (5) as follows:

$$z_{rj} \leq \alpha_r^U y_j \quad \forall j = 1, \ldots, m \tag{15}$$

Here we use $\alpha_r^U = x_i^{ref} - s\ d_i^L$, with s varying from zero to one and $\Delta\alpha_i^U$ is a parameter as above. Thus, running ALB_1^U repeatedly until the problem is either infeasible or it has a solution higher than the current upper bound for certain s^* one identifies new lower bound as follows $x_i^{ref} - s\ d_i^L$. In this case, one could start with $s = 0$ or with a value of s such that $x_i^{ref} - s\ d_i^L < f_x^{(2)}$ when $f_x^{(2)}$ is not used as x_r^{ref}.

The above presented bound contraction procedure can also be implemented for both variables of the bilinearity. We do not use it in this article, but we explain the details in the Appendix. In addition, the choice of the starting point ε and stepsize Δs is of importance to speed up the computations. For water management applications that we solved (see below), the method seems fast enough that improvements in solution time would be marginal. It is clear that a large starting point ε will have a better chance of eliminating a portion of the feasible space, but it will be sometimes smaller than what can really be eliminated. This is a conservative approach in which one repeats the same procedure (with a large ε) until bound contraction is exhausted. Likewise a larger stepsize Δs is used to start when an optimistic approach in which a larger portion of the feasible region is aimed at being eliminated. It is unclear if general rules exist to determine the most appropriate value for these parameters.

The algorithm then can proceed with this bound contraction until upper and lower bounds are close within a tolerance. If no further contraction can be made, the procedure needs to use a decomposition strategy of some sort where sub-problems are created. One such procedure could be a branch and bound scheme. We detailed such a procedure, where we perform contraction at each node in our other work (Faria & Bagajewicz, 2010c).

4. Global optimization algorithm

The bound contraction algorithm for contraction on one of the variables of the bilinear term is the following:

1. Assume that $\bar{x}_i^L = x_i^L, \bar{x}_i^U = x_i^U$
2. Run the LB model to get $\hat{z}_{ij}$, $\hat{x}_i$ and $\hat{y}_j$. Calculate x_i^{ref} and y_j^{ref}.

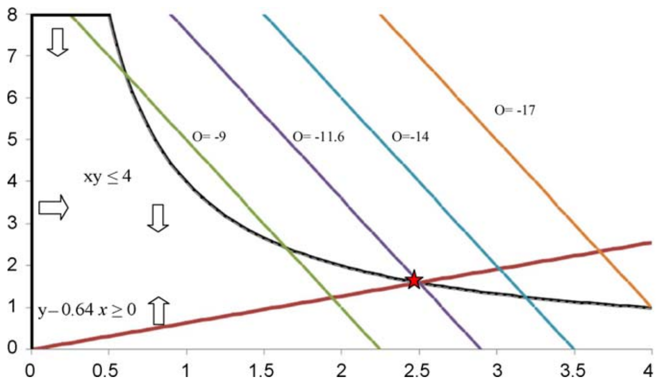

Fig. 2. Feasible region and optimum for example 1.

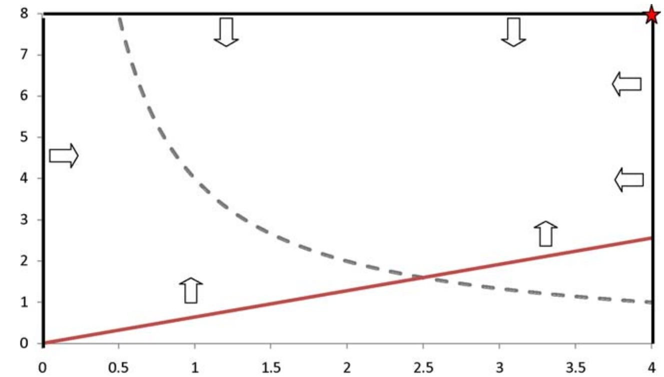

Fig. 3. Feasible region and optimum for Problem LB (example 1, 1st iteration).

3. Use $\hat{z}_{ij}$, $\hat{x}_i$ and $\hat{y}_j$ as initial values to calculate the UB by running the original MINLP. Alternatively, if this gives an infeasible answer, one can try some problem specific ad hoc upper bound versions of the problem.
4. Calculate all the distances d_i^L and d_i^U. Determine the variable r that has the smallest distance. If $d_r^L < d_r^U$ go to step 5. Otherwise, go to step 6.
5. Run problem ALB_r^L for different values of s until it is infeasible or it has an objective larger than the current upper bound of the problem. Set $\bar{x}_r^U \leftarrow \bar{x}_r^L + s^* \; d_r^U$. We use a starting point $s = \varepsilon$ and steps Δs added to it. Go to step 7.
6. Run problem ALB_r^U for different values of s until it is infeasible or it has an objective larger than the current upper bound of the problem. Set $\bar{x}_r^L \leftarrow \bar{x}_r^U - s^* \; d_r^L$. Go to step 7.
7. If $\bar{x}_i^U - \bar{x}_i^L < \varepsilon$ (the tolerance) for ALL $i \in I$ or (UB-LB)/UB < tolerance, then stop. Otherwise go to step 8.
8. If no variable was contracted in the previous pass, split the problem in sub-problems and repeat 1–7 for each sub-problem.

A simpler alternative is to go sequentially to each variable and try both sides running ALB_r^L and ALB_r^U, that is to bound contract from both sides without using any distance comparison. This is the technique we use in the illustrations.

5. Illustration-example 1

We illustrate the method using the following small example.

$$\left.\begin{array}{l} Min \quad O = -4x - y \\ s.t \\ xy \le 4 \\ y - 0.64x \ge 0 \\ x \in [0, 4] \;\; ; \;\; y \in [0, 8] \end{array}\right\} \qquad (16)$$

The feasible region of this problem is shown in Fig. 2. The optimum is at $\hat{x} = 2.5$, $\hat{y} = 1.6$ (indicated in the figure with the ★ symbol) with an objective value of $\hat{O} = -11.6$. The other local minimum is found at $\hat{x} = 0.5$, $\hat{y} = 8$ with an objective value of $\hat{O} = -10.0$.

Consider now that we contract on variable x. Then, when our lower bound model becomes:

$$\left.\begin{array}{l} LB = Min \quad O = -4x - y \\ s.t \\ z \le 4 \\ y - 0.64x \ge 0 \\ z \; \le \bar{x}^U y = 4y \\ z \; \ge \bar{x}^L y = 0 \\ x \in [0, 4] \;\; ; \;\; y \in [0, 8] \end{array}\right\} \qquad (17)$$

We notice that $z \le 4$ and $z \le 4y$ do not impose any restrictions on y and therefore, $y \in [0,8]$. The feasible region and the solution is shown in Fig. 3.

The solution is: $\hat{x} = 4.0$, $\hat{y} = 8$ (upper right corner of the feasible region) and objective $\hat{z} = 0$ with objective $\hat{O} = -24.0$. At this point we use this starting point in a local optimizer (we used GAMS/CONOPT) and we do not obtain the global minimum, but rather we obtain the other local minimum at $\hat{x} = 0.5$, $\hat{y} = 8$ with an objective value of $\hat{O} = -10.0$.

Quite clearly all reference values are the same, that is $x_i^{ref} = \hat{z}/\hat{y} = 0$. Thus $d^L = x^{ref} - \bar{x}^L = 0$ and $d_i^U = \bar{x}_i^U - x_i^{ref} = 4$. As a side note, we note also that we used $x_i^{ref} = \hat{z}/\hat{y}$ as the best estimate of x_i. We note, however, that in this case $\hat{x}$ is in fact better, but for illustration purposes we stick to x_i^{ref}. Thus, because x^{ref} is closer to $\bar{x}^L$ (in fact equal), then we use $\alpha^L = x_r^{ref} + s \; d^U = 4s$, and write problem ALB^L as follows:

$$\left.\begin{array}{l} \mathrm{ALB}^L = Min \quad O = -4x - y \\ s.t \\ z \le 4 \\ y - 0.64x \ge 0 \\ z \; \le \bar{x}^U y = 4y \\ z \; \ge \alpha^L y = 4s\,y \\ x \in [4s, 4] \;\; ; \;\; y \in [0, 8] \end{array}\right\} \qquad (18)$$

We notice that now, $z \le 4$, $z \le 4y$ and $z \ge 4sy$ do impose restrictions on y and therefore, $y \in [0,1/s]$ (we remind the reader that $s \in [0,1]$). The feasible region for different values of s and the corresponding solutions are shown in Fig. 4 we used $\varepsilon = 0.2$ and $\Delta s = 0.2$.

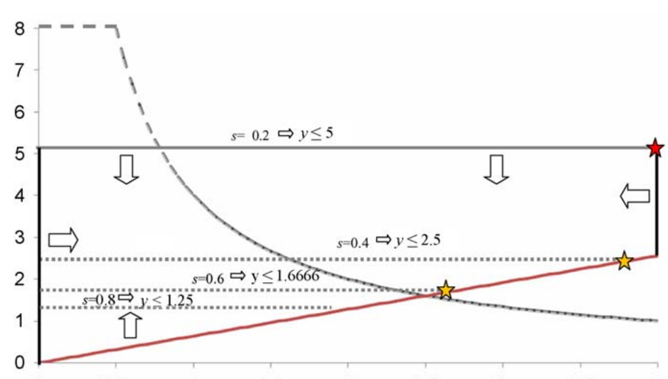

Fig. 4. Feasible region and optima for Problem ALB^L (example 1, 1st iteration).

Fig. 4 also illustrates the location of the optima. They are at [4,5] [3.90625,2.5] and [2.6040,1.6666] corresponding to values of s = 0.2, 0.4 and 0.6. The corresponding objectives are $O = -21, -18.125$, and -12.0833. For s = 0.8 the problem is infeasible. We note that the lower bound for x in this last problem is $x^L = 3.2$.

Based on the above results, we update the upper bound to be $\bar{x}^U \leftarrow \bar{x}_r^L + s^* d_r^U = 3.2$. One can also find a better bound by adding a bisection step, which would identify the value at which the lower bound is equal to the upper bound. This value is s = 0.725015. Had GAMS/CONOPT identified the global optimum as its UB, the value obtained by bisection would be s = 0.625.

At this time, without the bisection step, one resorts to calculate the new LB, solving the following problem:

$$\left.\begin{array}{l} LB = Min \quad O = -4x - y \\ s.t \\ z \le 4 \\ y - 0.64x \ge 0 \\ z \ \le \bar{x}^U y = 2.604y \\ z \ \ge \bar{x}^L y = 0 \\ x \in [0, 3.2] \ ; \ y \in [0, 8] \end{array}\right\} \tag{19}$$

The solution is: $\hat{x} = 3.2, \hat{y} = 8.0$ and $\hat{z} = 0$ with objective $\hat{O} = -18.416$. We now run the upper bound model (original problem), with this starting point and we obtain the global optimum, so the upper bound is updated to UB = −11.6.

With the above results $x_i^{ref} = \hat{z}/\hat{y} = 0$ and therefore $d^L = x^{ref} - \bar{x}^L = 0$ again and $d_i^U = \bar{x}_i^U - x_i^{ref} = 3.2$. Thus, because x^{ref} is closer to $\bar{x}^L$ (in fact equal again), then we use $\alpha^L = x_r^{ref} + s \ d^U = 3.2s$, and write problem ALB^L as follows:

$$\left.\begin{array}{l} \text{ALB}^L = Min \quad O = -4x - y \\ s.t \\ z \le 4 \\ y - 0.64x \ge 0 \\ z \ \le \bar{x}^U y = 3.2y \\ z \ \ge \alpha^L y = 3.2sy \\ x \in [3.2s, 3.2] \ ; \ y \in [0, 8] \end{array}\right\} \tag{20}$$

Again, $z \le 4$, $z \le 4y$ and $z \ge 3.2sy$ do impose restrictions on y and therefore, $y \in [0, 4/(3.2s)]$. We therefore run problem ALB^L using ε = 0.2 and Δs = 0.2. Fig. 5 depicts the feasible regions of the successive problems solved with increasing s.

Fig. 5 shows the location of the optima. They are at [3.2,6.25 [3.2,3.125] and [3.2,2.083] corresponding to values of s = 0.2, 0.4 and 0.6. The corresponding objectives are O = −19.05, −15.925, −14.883.

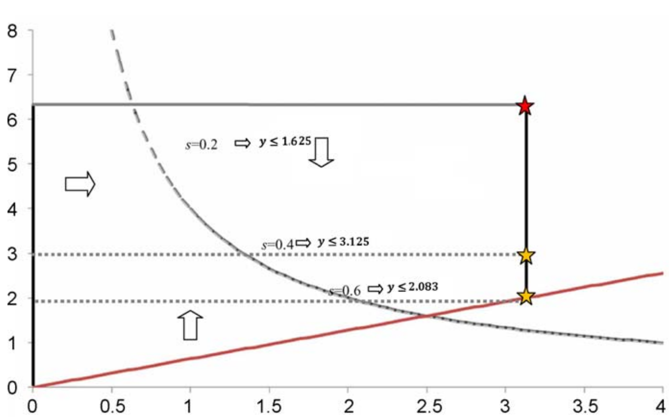

Fig. 5. Feasible region and optima for Problem ALB^L (example 1, 2nd iteration).

When s = 0.8 the problem is infeasible. The lower bound on x for the last infeasible problem was x^L = 2.56, which becomes the new upper bound for the next iteration. In the third iteration, one founds that no contraction is possible for the upper bound. Thus, we try to contract the lower bound by solving ALB_r^U. When this is done, one finds that for s = 0.8, the problem renders an objective value of −10.048, which is higher than the current upper bound. Therefore the lower bound is updated to be $\bar{x}_r^L \leftarrow \bar{x}_r^U - s^* d_r^U = 3.2 - 0.8 \times 2.56 = 0.512$. The process continues for four more iterations until the lower and upper bound are smaller than the tolerance.

6. Illustration using water management problems

We start with the very well known water management problem (see Bagajewicz, 2000).

The non-linear model to solve the water allocation problem is the following:

6.1. Water balance at the water-using units

$$\sum_w FWU_{w,u} + \sum_{u*} FUU_{u*,u} + \sum_r FRU_{r,u} = \sum_s FUS_{u,s} + \sum_{u*} FUU_{u,u*} + \sum_r FUR_{u,r} \forall u \tag{21}$$

where $FWU_{w,u}$ is the freshwater from the sources to the water using units, $FUU_{u*,u}$ is the water sent from process u^* to process u, $FRU_{r,u}$ is the water sent from the regeneration process r to water using unit u, $FUR_{u,r}$ is the water sent from the the water using process u to regeneration unit r, and $FUS_{u,s}$ is the water sent from water using unit u to sink s.

6.2. Water balance at the regeneration units

$$\sum_w FWR_{w,r} + \sum_u FUR_{u,r} + \sum_{r*} FRR_{r*,r} = \sum_s FRS_{r,s} + \sum_u FRU_{r,u} + \sum_{r*} FRR_{r,r*} \quad \forall r \tag{22}$$

where $FWR_{w,r}$ is the water sent from freshwater source w to the regenratiopn unit r, $FRR_{r*,r}$ is the water sent from one regeneration unit to another and $FRS_{r,s}$ the water sent from the regeneration unit r to sink s.

6.3. Contaminant balance at the water-using units

$$\left.\begin{array}{l} \sum_w CW_{w,c} FWU_{w,u} + \sum_{u*} FUU_{u*,u} CU_{u*,c} + \sum_r FRU_{r,u} CR_{r,c} + \Delta M_{u,c} = \\ = CU_{u,c} \left(\sum_{u*} FUU_{u,u*} + \sum_s FUS_{u,s} + \sum_r FUR_{u,r} \right) \quad \forall u, c \end{array}\right\} \tag{23}$$

where $CW_{w,c}$ is the concentration of contaminant c in freshwater w, $CU_{u*,c}$ is the concentration of contaminant c in the outlet of water using unit u, and $CR_{r,c}$ is the concentration of contaminant c at the outlet of regeneration unit r.

6.4. Contaminant balance at the regeneration processes

For contaminants that are not treated in the regeneration process, we write:

Table 1
Limiting data of example 2.

Process	Contaminant	Mass load (kg/h)	$C^{in,max}$ (ppm)	$C^{out,max}$ (ppm)
(1) Caustic treating	Salts	0.18	300	500
	Organics	1.2	50	500
	H_2S	0.75	5000	11,000
	Ammonia	0.1	1500	3,000
(2) Distillation	Salts	3.61	10	200
	Organics	100	1	4,000
	H_2S	0.25	0	500
	Ammonia	0.8	0	1,000
(3) Amine sweetening	Salts	0.6	10	1,000
	Organics	30	1	3,500
	H_2S	1.5	0	2,000
	Ammonia	1	0	3,500
(4) Merox-I sweetening	Salts	2	100	400
	Organics	60	200	6,000
	H_2S	0.8	50	2,000
	Ammonia	1	1000	3,500
(5) Hydrotreating	Salts	3.8	85	350
	Organics	45	200	1,800
	H_2S	1.1	300	6,500
	Ammonia	2	200	1,000
(6) Desalting	Salts	120	1000	9,500
	Organics	480	1000	6,500
	H_2S	1.5	150	450
	Ammonia	0	200	400

$$\left.\begin{aligned}&\sum_{w} CW_{w,c}\ FWR_{w,r} + \sum_{u} FUR_{u,r} CU_{u,c} + \sum_{r*} FRR_{r*,r} CR_{r*,c} = \\ &= CR_{r,c}\left(\sum_{s} FRS_{r,s} + \sum_{u} FRU_{r,u} + \sum_{r*} FRR_{r,r*}\right) \quad \forall r, c\end{aligned}\right\} \tag{24}$$

where $CW_{w,c}$ is concentration of contaminant c in the wastewater source and $CR_{r,c}$ is the concentration of contaminant c at the outlet of regeneration process r.

Regeneration processes are usually modeled using fixed concentrations at the outlet or fixed removal. For the cases in which the regeneration process is assumed to have fixed rate of removal, we use:

$$\left.\begin{aligned}&\left\{\sum_{w} CW_{w,c}\ FWR_{w,r} + \sum_{u} FUR_{u,r} CU_{u,c} + \sum_{r*} FRR_{r*,r} CR_{r*,c}\right\}(1 - RR_{r,c}) = \\ &= CR_{r,c}\left(\sum_{s} FRS_{r,s} + \sum_{u} FRU_{r,u} + \sum_{r*} FRR_{r,r*}\right) \quad \forall r, c\end{aligned}\right\} \tag{25}$$

where $RR_{r,c}$ is a given rate of removal of contaminant c in regeneration process r.

6.5. Maximum inlet concentration at the water-using units

$$\left.\begin{aligned}&\sum_{w} CW_{w,c} FWU_{w,u} + \sum_{u*} FUU_{u*,u} CU_{u*,c} + \sum_{r} FRU_{r,u} CR_{r,c} \le \\ &C_{u,c}^{in,\max}\left(\sum_{w} FWU_{w,u} + \sum_{u*} FUU_{u*,u} + \sum_{r} FRU_{r,u}\right) \forall u, c\end{aligned}\right\} \tag{26}$$

where $C_{u,c}^{in,\max}$ is the maximum inlet concentration in the water using units.

6.6. Maximum outlet concentration at the water-using units

$$CU_{u,c} \le C_{u,c}^{out,\max} \quad \forall u, c \tag{27}$$

where $C_{u,c}^{out,\max}$ is the maximum inlet concentration in the water using units.

6.7. Maximum outlet concentration at the sinks

$$\begin{aligned}&\sum_{u} FUS_{u,s} CU_{u,c} + \sum_{r} FRS_{r,s} CR_{r,c} \\ &\le CS_{s,c}^{out,\max}\left(\sum_{u} FUS_{s,u} + \sum_{r} FRS_{r,s}\right) \quad \forall s, c\end{aligned} \tag{28}$$

6.8. Existence of connections

These constraints are only necessary when one wants to minimize cost or there are restrictions in the connections capacities. When the target is minimum freshwater consumption and no connections limits are applied, they are no longer needed.

$$FWU_{w,u} \le F\max\ YWU_{w,u} \quad \forall w, u \tag{29}$$

$$FUU_{u,u*} \le F\max\ YUU_{u,u*} \quad \forall u, u* \tag{30}$$

$$FUS_{u,s} \le F\max\ YUS_{u,s} \quad \forall u, s \tag{31}$$

$$FUR_{u,r} \le F\max\ YUR_{u,r} \quad \forall r, u \tag{32}$$

$$FRU_{r,u} \le F\max\ YRU_{r,u} \quad \forall r, u \tag{33}$$

$$FRS_{r,s} \le F\max\ YRS_{r,s} \quad \forall r, s \tag{34}$$

$$FRR_{r,r*} \le F\max\ YRR_{r,r*} \quad \forall r, r* \tag{35}$$

where $YWU_{w,u}$, $YUU_{u,u*}$, $YUS_{u,s}$, $YUR_{u,r}$, $YRU_{r,u}$, $YRS_{r,s}$, $YRR_{r,r*}$ are binary variables that are zero if no flow is going through the connection.

6.9. Minimum flowrates

Minimum flowrate constraints are optional constraints that are introduced when small flowrates are undesired.

$$FWU_{w,u} \geq F\min\ YWU_{w,u} \quad \forall w, u \tag{36}$$

$$FUU_{u,u*} \geq F\min\ YUU_{u,u*} \quad \forall u, u* \tag{37}$$

$$FUS_{u,s} \geq F\min\ YUS_{u,s} \quad \forall u, s \tag{38}$$

$$FUR_{u,r} \geq F\min\ YUR_{u,r} \quad \forall r, u \tag{39}$$

$$FRU_{r,u} \geq F\min\ YRU_{r,u} \quad \forall r, u \tag{40}$$

$$FRS_{r,s} \geq F\min\ YRS_{r,s} \quad \forall r, s \tag{41}$$

$$FRR_{r,r*} \geq F\min\ YRR_{r,r*} \quad \forall r, r* \tag{42}$$

6.10. Freshwater consumption—objective function

$$consu = \sum_{w}\left(\sum_{u} FWU_{w,u} + \sum_{r} FWR_{w,r}\right) \tag{43}$$

6.11. Total cost—objective function

$$TotalCost = OperatingCost + df\ CapitalCost \tag{44}$$

6.12. Operating cost

$$\begin{aligned} OperatingCost = {} & \sum_{w} fwc_{w}\left(\sum_{u} FWU_{w,u} + \sum_{r} FWR_{w,r}\right) \\ & + \sum_{r} \mathrm{RegOC}\left(\sum_{w} FWR_{w,r} + \sum_{u} FUR_{u,r} + \sum_{r*} FRR_{r*,r}\right) \end{aligned} \tag{45}$$

6.13. Capital cost

$$\begin{aligned} CapitalCost = {} & +\sum_{r} RegCC\left(\sum_{w} FWR_{w,r} + \sum_{u} FUR_{u,r} + \sum_{r*} FRR_{r*,r}\right)^{\gamma} + \\ & +\sum_{u}\left(\begin{array}{l} \sum_{w}\left(LWU_{w,u} * \left(\beta * YWU_{w,u} + \alpha * FWU_{w,u}\right)\right) \\ +\sum_{u*}\left(LUU_{u,u*} * \left(\beta * YUU_{u,u*} + \alpha * FUU_{u,u*}\right)\right) \\ +\sum_{r}\left(LUR_{u,r} * \left(\beta * YUR_{u,r} + \alpha * FUR_{u,r}\right)\right) \\ +\sum_{s}\left(LUS_{u,s} * \left(\beta * YUS_{u,s} + \alpha * FUS_{u,s}\right)\right) \end{array}\right) \\ & +\sum_{r}\left(\begin{array}{l} \sum_{w}\left(LWR_{w,r} * \left(\beta * YWR_{w,r} + \alpha * FWR_{w,r}\right)\right) \\ +\sum_{u}\left(LRU_{r,u} * \left(\beta * YRU_{r,u} + \alpha * FRU_{r,u}\right)\right) \\ +\sum_{r*}\left(LRR_{r,r*} * \left(\beta * YRR_{r,r*} + \alpha * FRR_{r,r*}\right)\right) \\ +\sum_{s}\left(LRS_{u,s} * \left(\beta * YRS_{r,s} + \alpha * FRS_{r,s}\right)\right) \end{array}\right) \end{aligned} \tag{46}$$

7. Example 2—minimum freshwater consumption (MINLP)

This example is the refinery case presented by Koppol, Bagajewicz, Dericks, and Savelski (2003), which has four key contaminants (salts, H_2S, Organics and ammonia), six water-using units and three regeneration processes. The limiting data of the water-using units are shown in Table 1.

This network without reuse consumes 144.8 t/h of freshwater and the objective is to minimize freshwater use. No integers are included, so the problem is non-convex NLP.

The potential regeneration processes are modeled as processes with fixed outlet concentrations. They are reverse osmosis, which reduces salts to 20 ppm; API separator followed by ACA, which reduces organics to 50 ppm; and, Chevron wastewater treatment,

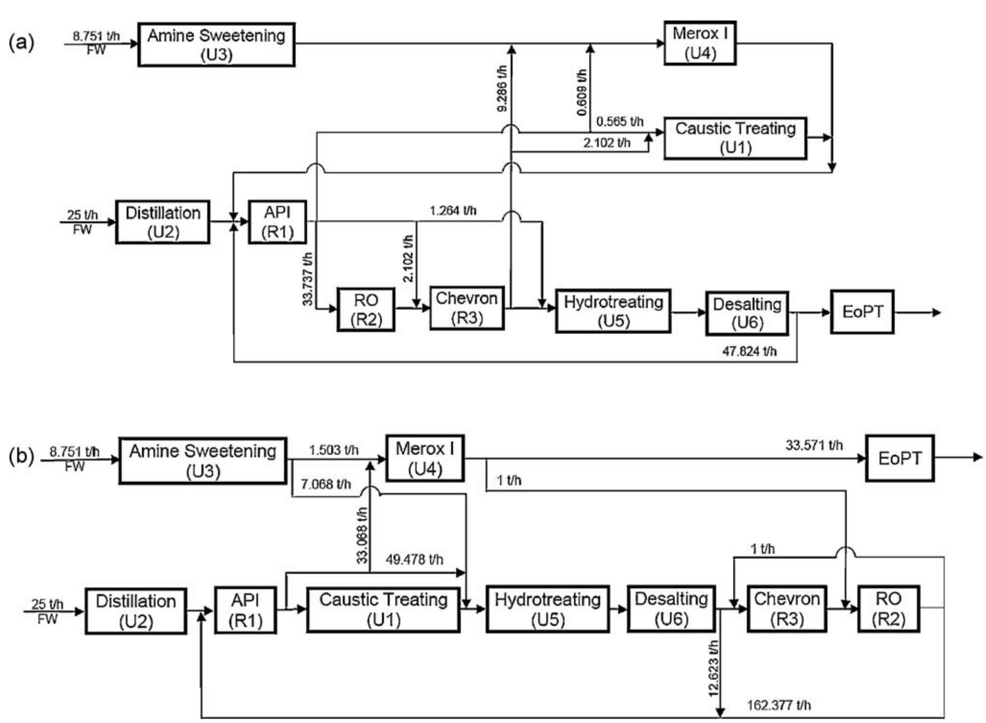

Fig. 6. Optimum network of example 1. (a) Koppol et al. (2003) and (b) ours.

Table 2
Water using units limiting data of example 3.

Process	Contaminant	Mass load (kg/h)	$C^{in,max}$ (ppm)	F_{max} (t/h)
1	A	1	0	40
	B	1.5	0	
	C	1	0	
2	A	1	50	50
	B	1	50	
	C	1	50	
3	A	1	50	50
	B	1	50	
	C	1	50	
4	A	2	50	50
	B	2	50	
	C	2	50	
5	A	1	25	25
	B	1	25	
	C	0	25	

Table 3
Regeneration processes data of example 3.

Process	Contaminant	Removal ratio (%)	OPN_r	CCR_r
1	A	95	1	16,800
	B	0		
	C	0		
2	A	0	0.04	9,500
	B	0		
	C	95		
3	A	0	0.0067	12,600
	B	95		
	C	0		

which reduces H_2S to 5 ppm and ammonia to 30 ppm. The optimum solution obtained by Koppol et al. (2003) reaches a minimum freshwater consumption of 33.571 t/h. Here we solve the problem as an MINLP one because we do not allow the existence of small flowrates through connections. We contracted on the water using

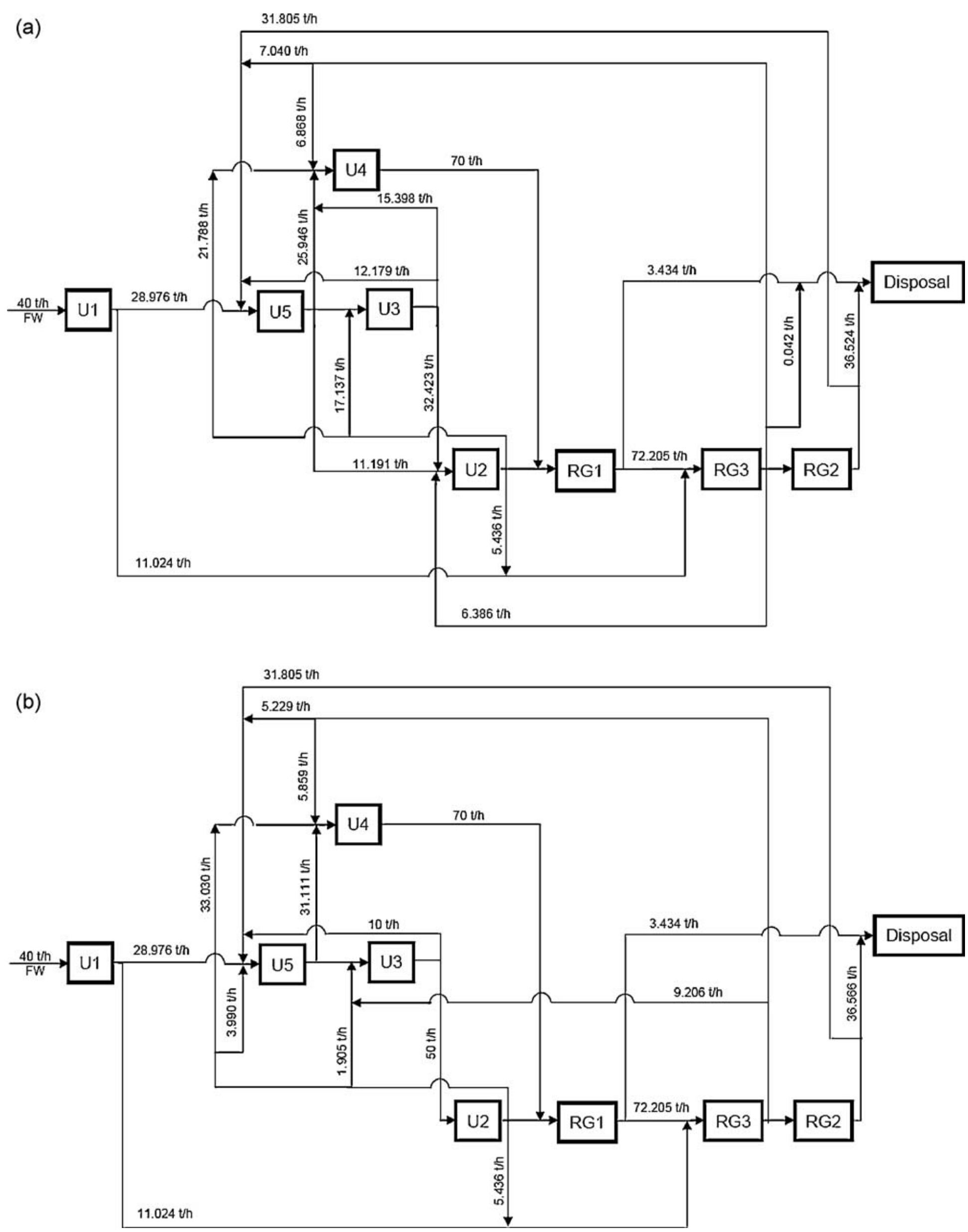

Fig. 7. Optimum network of example 3. (a) NLP problem and (b) MINLP problem with flow restrictions.

units and regeneration process flowrates only and used $\varepsilon = 0$ and $\Delta s = 0.45$. The solving time for this problem is 41 CPUs. We used and Intel Core 2 Duo processor (2 GHz and 2 GB of RAM), CPLEX 10.1 and DICOPT (CONOPT/CPLEX) all part of GAMS 22.3.

The difficulty of this problem when solved as an MINLP is that there are several local minima that the local solver can find. Although the lower bound value at the root node coincides with the global solution, a feasible solution (upper bound) at minimum consumption is only found after the procedure generates starting point for the MINLP solver from different parts of the feasible region.

Fig. 6 shows the networks obtained by Koppol et al. (2003), which was solved as an NLP problem, and ours, which forbids the existence of small flowrates through the connections.

8. Example 3—minimum TAC

This example involves three contaminants and has five water-using units with fixed flowrates and three regeneration processes, which is the largest system presented by Karuppiah and Grossmann (2006). The data for this example is presented below (Tables 2 and 3). The discharge limit of all the contaminants is 10 ppm. The cost of freshwater is \$1/t, the annualized factor is 0.1 and the plant runs 8000 h/year. In this case, for the cost, we use piece-wise linear underestimators for the concave terms in the objective function when we run the lower bound models.

We found the optimum solution (\$1,033,810.95/year) in 6.2 CPU seconds. We only applied the bound contraction procedure to the flowrates through the regeneration processes and used $\varepsilon = 0.1$ and $\Delta s = 0.45$. The procedure finds the solution within the first iteration, which significantly contracts the flowrates through the regeneration processes and brings the lower bound objective function from \$1,016955 to \$1,023,546 which has a 0.99% gap. The solution is presented in Fig. 7 and is the same as the one obtained by Karuppiah and Grossmann (2006).

To avoid small flowrates we added a limitation that, if the flowrate exists, it has to have a flowrate larger or equal 1 t/h. This addition turns the model into an MINLP model. In this paper, our solution bound contracts the flowrates of regeneration processes using $\varepsilon = 0.1$ and no increments option was obtained in 50.22 s. The minimum TAC was found in the first iteration as being \$1,033,832.

9. Conclusions

We presented a bound contraction-based global optimization algorithm. Unlike several other algorithms that construct lower bound by partitioning key variables into several intervals, thus requiring binary variables to identify in which interval the lower bound solution lies, our algorithm performs bound contraction without the need to use additional binary variables. The lower bound is run after successive bound contraction steps until the gap between the lower bound and the upper bound reaches a specific threshold.

Acknowledgement

Débora Faria acknowledges support from the CAPES/Fulbright Program (Brazil).

Appendix A. Appendix

A.1. Extended bound contraction procedure

The above bound contraction algorithm can also be run when both variables are involved in the procedure. We present now this extended bound contraction notion. In this case, for the lower bound, Eq. (1) is substituted by Eqs. (4) and (5) as shown above plus the following two constraints:

$$z_{ij} \geq x_i \bar{y}_j^L \quad \forall i = 1, \ldots, n; \quad \forall j = 1, \ldots, m \tag{47}$$

$$z_{ij} \leq x_i \bar{y}_j^U \quad \forall i = 1, \ldots, n; \quad \forall j = 1, \ldots, m \tag{A.2}$$

where we use updated bounds for y_j ($\bar{y}_j^L$ and $\bar{y}_j^U$).

Once this LB model is solved we define reference values for x_i (x_i^{ref}) as above, and we also define distances for y_j (y_j^{ref}) as follows:

$$y_j^{ref} = f_y^{(i)}(\hat{z}_{i1}, \hat{z}_{i2}, \ldots, \hat{z}_{im}; \hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n) \quad \forall j = 1, \ldots, m \tag{48}$$

We use the same options for $f(\bullet)$, namely:

$$f_y^{(1)}(\hat{z}_{i1}, \hat{z}_{i2}, \ldots, \hat{z}_{im}; \hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n) = \frac{\sum_{i=1,\ldots n} \hat{z}_{ij}}{\sum_{i=1,\ldots n} \hat{x}_j} \quad \forall j = 1, \ldots, m \tag{A.4}$$

$$f_y^{(2)}(\hat{z}_{i1}, \hat{z}_{i2}, \ldots, \hat{z}_{im}; \hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n) = \underset{\forall i=1,\ldots n}{Max} \left\{ \frac{\hat{z}_{ij}}{\hat{x}_i} \right\} \quad \forall j = 1, \ldots, m \tag{A.5}$$

$$f_x^{(3)}(\hat{z}_{i1}, \hat{z}_{i2}, \ldots, \hat{z}_{im}; \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m) = \underset{\forall j=1,\ldots m}{Min} \left\{ \frac{\hat{z}_{ij}}{\hat{y}_j} \right\} \quad \forall i = 1, \ldots, n \tag{A.6}$$

We also define the same distances and the algorithm is run exactly as described above, except that all variables of the bilinear terms are considered for contraction. In addition, the presence of both variables as candidates for contraction may prompt the addition of some ad hoc problem specific.

The advantages of performing bound contractions in both variables are obvious: more variables are contracted in any single iteration. There are however some possible disadvantages, like the fact that more time is spent contracting in each iteration. The consequences on computational time are unclear and likely depend on the problem.

References

Adhya, N., Tawarmalani, M., & Sahinidis, N. V. (1999). A Lagrangian approach to the pooling problem. *Industrial Engineering and Chemistry Research*, *38*, 1956–1972.

Androulakis, P., Maranas, C. D., & Floudas, C. A. (1995). αBB: A global optimization method for general constrained nonconvex problems. *Journal of Global Optimization*, *7*(4), 337–363.

Bagajewicz, M. J. (2000). A review of recent design procedures for water networks in refineries and process plants. *Computer and Chemical Engineering*, *24*, 2093.

Ben-Tal, A., Eiger, G., & Gershovitz, V. (1994). Global minimization by reducing the duality gap. *Mathematical Programming*, *63*, 193–212.

Bergamini, M. L., Aguirre, P., & Grossmann, I. (2005). Logic based outer approximation for global optimization of synthesis of process networks. *Computer and Chemical Engineering*, *29*, 1914.

Bergamini, M. L., Grossmann, I., Scenna, N., & Aguirre, P. (2008). An improved piecewise outer-approximation algorithm for the global optimization of MINLP models involving concave and bilinear terms. *Computer and Chemical Engineering*, *32*, 477–493.

Faria, D. C., & Bagajewicz, M. J. (2006). Retrofit of water networks in process plants. In *Proceedings of the XXII interamerican congress of chemical engineering* Buenos Aires, Argentina.

Faria, D. C., & Bagajewicz, M. J. (2008). A new approach for the design of multicomponent water/wastewater networks. *Computer Aided Chemical Engineering*, *25*, 43–48.

Faria, D. C., & Bagajewicz, M. J. (2009). Profit-based grassroots design and retrofit of water networks in process plants. *Computers and Chemical Engineering*, *33*(2), 436–453.

Faria, D. C., & Bagajewicz, M. J. (2010a). On the appropriate modeling of the process plant water systems. *AIChE Journal*, *56*(3), 668–689.

Faria, D. C., & Bagajewicz, M. (2010b). On the degeneracy of the water/wastewater allocation problem in process plants. *Industrial and Engineering Chemistry Research*, *49*(9), 4340–4351.

Faria, D. C., & Bagajewicz, M.J. (2010c). A new approach for global optimization of a class of MINLP problems with applications to water management and pooling problems. AIChE Journal. (To appear).

Floudas, C. A. (2000a). *Deterministic global optimization: Theory, methods and applications. Nonconvex optimization and its applications*. Kluwer Academic Publishers.

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

Floudas, C. A., Akrotirianakis, I. G., Caratzoulas, S., Meyer, C. A., & Kallrath, J. (2005). Global optimization in the 21st century: Advances and challenges. *Computers & Chemical Engineering*, *29*(6), 1185–1202.

Floudas, C. A., & Gounaris, C. E. (2009). A review of recent advances in globaloptimization. *Journal of Global Optimization*, *45*, 3–38.

Hansen, E. R. (1979). Global optimization using interval analysis: The one-dimensional case. *Journal of Optimization Theory and Applications*, *29*, 331–344.

Hansen, E., & Walster, G. W. (2004). *Global optimization using interval analysis. Pure and applied mathematics*. Marcel Dekker.

Horst, R., & Tuy, H. (2003). *Global optimization: Deterministic approaches*. Springer.

Karuppiah, R., & Grossmann, I. E. (2006a). Global optimization for the synthesis of integrated water systems in chemical processes. *Computer and Chemical Engineering*, *30*, 650–673.

Karuppiah, R., & Grossmann, I. E. (2006b). Global optimization of multiscenario mixed integer nonlinear programming models arising in the synthesis of integrated water networks under uncertainty. *Computer Aided Chemical Engineering*, *21–2*, 1747–1752.

Karrupiah, R., & Grossmann, I. E. (2008). A Lagrangian based branch-and-cut algorithm for global optimization of nonconvex mixed-integer nonlinear programs with decomposable structures. *Journal of Global Optimization*, *41*, 1573–2916.

Koppol, A. P. R., Bagajewicz, J. M., Dericks, B. J., & Savelski, M. J. (2003). On zero water discharge solutions in process industry. *Advances in Environmental Research*, *8*, 151–171.

Kuno, T., & Utsunomiya, T. (2000). A Lagrangian based branch-and-bound algorithm for production–transportation problems. *Journal of Global Optimization*, *18*, 59–73.

McCormick, G. P. (1976). Computability of global solutions to factorable nonconvex programs. *Mathematical Programming*, *10*, 146–175.

Meyer, C. A., & Floudas, C. A. (2006). Global optimization of a combinatorially complex generalized pooling problem. *AIChE Journal*, *52–3*, 1027–1037.

Moore, R. E. (1966). *Interval analysis*. Prentice-Hall.

Moore, R., Hansen, E. R., & Leclerc, A. (1992). Rigorous methods for global optimization. In C. A. Floudas, & P. M. Pardalos (Eds.), *Recent advances in global optimization*. Princeton: Princeton University Press, p. 321.

Pardalos, P. M., Romeijn, H. E., & Tuy, H. (2000). Recent developments and trends in global optimization. *Journal of Computational and Applied Mathematics*, *124*(1–2), 209–228.

Ratschek, H., & Rokne, J. (1988). *New computer methods for global optimization*. New York, NY: Halsted Press.

Ruiz, J. P., & Grossmann, I. E. (2010). Strengthening of lower bounds in the global optimization of bilinear and concave generalized disjunctive programs. *Computers and Chemical Engineering*, *34*(6), 914–930.

Sahinidis, N. V. (1996). BARON: A general purpose global optimization software package. *Journal of Global Optimization*, *7*(4), 337–363.

Savelski, M., & Bagajewicz, M. (2001). On the use of linear models for the design of water utilization systems in refineries and process plants. *Chemical Engineering Research & Design*, *79*, 600–610.

Sherali, H. D., & Adams, W. P. (1999). *A reformulation-linearization technique for solving discrete and coninuous nonconvex problems. Nonconvex optimization and its applications*. Kluwer Academic Publishers.

Tawarmalani, M., & Sahinidis, N. V. (2002). *Convexification and global optimization in continuous and mixed-integer nonlinear programming: Theory, algorithms, software, and applications nonconvex optimization and its applications*. Kluwer Academic Publishers.

Vaidyanathan, R., & El-Halwagi, M. (1994). Global optimisation of nonconvex programs via interval analysis. *Computers & Chemical Engineering*, *18*, 889–897.

Zamora, J. M., & Grossmann, I. E. (1999). A branch and contract algorithm for problems with concave univariate, bilinear and linear fractional terms. *Journal of Global Optimization*, *14*, 217–249.