A NEW APPROACH FOR GLOBAL OPTIMIZATION OF MINLP PROBLEMS WITH BILINEAR AND CONCAVE UNIVARIATE TERMS - PART I: THEORY

Débora C. Faria and Miguel J. Bagajewicz*

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

Keywords: Mathematical Programming, Optimization, Global Optimization

(*) Corresponding author

Abstract

One of the biggest challenges in solving optimization engineering problems are rooted in the nonlinearities and non-convexities, which arise from bilinear terms corresponding to component material balances and/or concave functions used to estimate capital cost of equipments. The procedure we propose uses an MILP lower bound constructed using discretization of certain variables, similar to the one used by other approaches. The core of the method is to bound contract a set of variables that are not necessarily the ones being discretized. The procedure for bound contraction consists of a novel interval elimination procedure that has several variants. Once bound contraction is exhausted the method increases the number of intervals or resorts to a branch and bound strategy where bound contraction takes place at each node. Our procedure shows being more efficient than others at least for the class of problems tested. The illustrations (Part II) focus on two MINLP problems: water management and pooling problems.

1. Introduction

Several methods have been proposed for global optimization, many of which became popular in the chemical engineering community, some having reached commercial status, like BARON (Sahinidis, 1996), COCOS, GlobSol, ICOS, LGO, LINGO, OQNLP, Premium Solver, or others that are well-known like the αBB (Androulakis et al., 1995). More comprehensive understanding of evolution and advances of global optimization can be found in several books (Sherali and Adams, 1999; Floudas, 2000a; Tawarmalani and Sahinidis, 2002; Horst and Tuy, 2003; Hansen and Walster, 2004) and recent paper reviews (Floudas 2000b; Pardalos et al., 2000; Floudas et al., 2005; Floudas and Gounaris, 2009).

The type of problems we address is those that contain bilinear terms where a flowrate and a concentration participate in bilinear terms, as part of component balances in processes. In addition, we consider the presence of univariate concave terms that are part of the objective function, typically representing equipment costs. Water management problems as well as pooling problems are such problems. We do not rule out extensions to other bilinear problems, which are part of future work. To approach this problem, we focus on logic-based procedures.

There are a few different approaches for global optimization. Because this is not a review and because we focus on a special type of bilinear problems, we only cite recent academic efforts that are close to the methodology we present and the specific optimization problems we target. Briefly, Lagrangean-based approaches (Ben-Tal et al., 1994; Adhya et al., 1999; Kuno and Utsunomiya, 2000; Karuppiah and Grossmann, 2008), and disjunctive programming-based methods (Ruiz and Grossmann, 2009) have been used. In turn, several global optimization procedures to handle bilinear and concave terms that are based on interval analysis have been presented: Intervals analysis arithmetic was first presented by Moore (1966) and then several subsequent paper tried to develop and improve global optimization approaches using interval analysis. Later, Vaidyanathan and El-Halwagi (1994) proposed an algorithm based on deleting infeasible subspaces using different tools to accelerate existing interval-based methods (Hansen, 1979; Ratschek and Rokne, 1988; Moore et al., 1992).

Another important class of methods are those based on branch and bound on key variables. Zamora and Grossmann (1999), presented a method for problems containing bilinear and fractional terms as well as univariate concave functions where they use such a branch and bound method but also performing variable bounds contractions at each node. For their lower bound model, they make use of a lower bound problem constructed using McCormick-type underestimators for the bilinear and fractional terms and linear underestimators for the concave functions. The set of complicating variables on which the branching is performed is chosen as the one that shows the largest deviation from the corresponding bilinear term, fractional term or the concave function. The upper bound solution, together with a special LP (or convex NLP) contraction sub-problem helps finding what bounds can be contracted. These new bounds are found by maximizing/minimizing each of their values subject to the linearized objective being smaller than the current UB.

They also use new bounding inequalities generated by the Lagrangean multipliers of the contraction sub-problem to improve the LBs in the branch and bound procedure.

To address bilinear terms in generalized pooling problems, Meyer and Floudas (2006) proposed a piece-wise reformulation-linearization technique (RLT). Reformulation consists of obtaining new redundant nonlinear constraints that are obtained by multiplying groups of valid constraints from the original problem. These new constraints are, of course, redundant in the original problem, but may be non-redundant in a convex relaxation. They partition the continuous space of one of the variables participating in a bilinear term in several intervals to generate a MINLP, alowing them to linearize the model to be able to generate lower bounds. Thus, linearization involves substituting bilinearities by a new variable and adding new constraints obtained by multiplying the bound inequalities (RLT). They suggest that a LB/UB scheme through which a gap can be reduced can be based in three alternatives: branching on the continuous variable, branching on the integers corresponding to the discretized intervals and a third one that they adopt: augmenting the LB problem with a set of binary variables to model a partition of the continuous space, and then reformulating and linearizing the problem as an MILP. The method does not involve a search for upper bounds and as they state, "several attempts and reformulations before a solution can be validated" are needed. Thus, the method is used just to verify the gap relative to the best known optimum solution and no procedure is presented to reduce the gap between lower and upper bounds. Different numbers of partitions of the continuous variables are considered to obtain the best lower bound. The method is able to generate very tight lower bounds at a cost of significant computational efforts due to the increase in numbers of binary variables.

Karuppiah and Grossmann (2006a), use a deterministic spatial branch and contract algorithm. To obtain a lower bound for the original NLP model, the bilinear terms are relaxed using the convex and concave envelopes (McCormick, 1976) and the concave terms of the objective function are replaced by underestimators generated by the secant of the concave term. To improve the tightness of the lower bound, piece-wise underestimators generated from partitioning of the flow variables are used to construct tighter envelopes and concave underestimators. The model is solved using disjunctions. They also choose the variable of the bilinear term that participates in the larger number of constraints to reduce the number of disjunctions. Logical cuts are also included to aid in the convergence (for example if two flows are to be identical, then they should fall within the same interval). Their algorithm then follows a bound contraction procedure which is a simplified version of the one used by Zamora and Grossmann (1999). As in Meyer and Floudas (2006), the number of partitions can make the lower bound tighter, but extra computational effort is needed. In a second paper, Karuppiah and Grossmann (2006b) extended the previous method to solve the multi-scenario case of the integrated water systems. In both cases, the relaxed model, which renders a lower bound, is used in a LB/UB framework. In the first case (Karuppiah and Grossmann, 2006a) a spatial branch and bound procedure is used. For the latter multi-scenario case (Karuppiah and Grossmann, 2006b), a spatial branch and cut algorithm is applied. The cuts are generated using a decomposition based on Lagrangean relaxation.

Bergamini et al. (2005) proposed an outer approximation method (OA) for global optimization; improvements followed later (Bergamini et al., 2008). The major modifications are related to a new formulation of the underestimators (which replace the concave and bilinear terms) using the delta-method of piecewise functions (see Padberd, 2000); and, the replacement of the most expensive step (global solution of the bounding problem) by a strategy based on the mathematical structure of the problem, which searches for better feasible solutions of fixed network structures. The improved outer approximation method relies in three sub-problems that need to be solved to feasibility instead to optimality. In turn, the model always look for solutions that are strictly lower (using a tolerance) than the current optimum solution.

In this paper we present a discretization methodology to obtain lower bounds and a new bound contraction procedure. Our lower bound model uses some modified versions of well-known over and underestimators (some of which used in the literature review above), to obtain MILP models. Our procedure differs from previous approaches based on LB/UB schemes in that it does not use a branch and bound methodology. Instead, we first discretize certain variables into several intervals and then use a bound contraction procedure directly using an interval elimination strategy. Conceptually, the technique works for a sufficiently high number of intervals, but if that becomes computationally too expensive, we allow a B&B to be used.

Although introducing logical cuts and/or performing bound contraction using interval arithmetic, either as a preliminary step or after each contraction, the method does not necessarily require introducing these logical cuts, reformulations or interval arithmetic-based contraction.

The paper is organized as follows: We present the solution strategy first, followed by a description of the discretization model and the lower bound MILP models. Then, the bound contraction procedure, as well as the auxiliary B&B procedures are described and illustrated. We then discuss similarities and differences with other methods in more detail. Finally, examples are presented and discussed.

2. Solution Strategy

After discretizing one of the variables in the bilinear terms, our method consists of a bound contraction step that uses a procedure for eliminating intervals. Once the bound contraction is exhausted, the method relies on increasing the number of intervals, or on a branch and bound strategy where the interval elimination takes place at each node. The discretization methodology (outlined below), generates linear models that guarantee to be lower bounds of the problem. Upper bounds are needed for the bound contraction procedure. These upper bounds can be usually obtained using the original MINLP model often initialized by the results of the lower bound model, although upper bounds can sometimes also be obtained using linear models.

Before we outline the strategy, we define variables:

- *Discretizing Variables:* These are the variables that are discretized into intervals and used to construct linerar relaxations of bilinear terms. The resulting models are MILP.
- *Bound Contracted Variables:* These are the variables that are discretized into intervals, only for the purpose of performing their bound contraction. The lower bound model will simply identify the interval in which the variable to be bound contracted lies and use this information in the elimination procedure. Clearly, these variables need not be the same as the discretizing variables.
- *Branch and Bound Variables:* These are the variables for which a branch and bound procedure is tried. It need not be the same set as the other two variables.

For example, in water management problems the bilinear terms are composed of the product of flowrates and concentrations. Thus, one can have a problem in which the discretizing variables are all or part of the concentrations, the bound contracted variables, be the flowrates and the B&B variables the flowrates as well. As we discuss below, the B&B is more efficient when the variables used are different from the discretizing variables when using McCormick's envelopes, which has information of the non discretized variable. Alternatively, one can use concentrations for both the discretizing and BC variables, with flowrates for B&B, or the discretizing variables could be both flowrates and concentrations (in which case the LB model is more efficient), the BC variables as well as the B&B variables the flowrates or the concentrations or both, and so on.

We also note that although the bound contract variable and branch and bound variable do not need to be the same as the discretized one, it is normal to have them being bound contracted or branched, as opposed to picking other variables. In some cases, picking the variable to bound contract different form the one to discretize renders tighter lower bounds as bound contraction takes place. However, we point out that the feasible region of the lower bound model can only become close to the feasible region of the upper bound when the discretized variables have discrete values within an ε tolerance and this can only be done through bound contraction and/or using branch and bound.

The global optimization strategy is now summarized as follows:

- Construct a lower bound model discretizing bilinear and quadratic terms, relaxing the bilinear terms as well as adding piece-wise linear underestimators of concave terms of the objective function. If the concave terms are not part of the objective function, then overestimators can be used, but this is not included in our current paper.
- The lower bound model is run identifying certain intervals as containing the solution for specific variables that are to be bound contracted. These variables need not be the same variables as the ones using to construct the lower bound. .
- Based on this information the value of the upper bound found by running the original MINLP using the information obtained by solving the lower bound model to obtain a good starting point. Other ad-hoc upper bounds can be constructed.

- A strategy based on the successive running of lower bounds where certain intervals are temporarily forbidden is used to eliminate regions of the feasible space. This is the bound contraction part.
- 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 discretization of the variables to a level in which the discrete sizes 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 sub-problems using a strategy such as the ones based on branch and bound procedure.

The first option of increasing discretization will not lead to further improvement in bound contraction if degenerate solutions (or very close to the global solutions) exist for different values of the discrete variables.

We discuss all these steps in the next few sections.

3. Discretization Methodology

We show here two different discretization strategies. The proposed approach consists of discretizing one of the variables of the bilinear terms, but one could also discretize both.

3.1 Bilinear Terms

There are different ways to linearize the bilinear terms using discrete points of one (or both) given variable(s). We use:

- *Direct Discretization* (our nomenclature). Some details of this technique were presented earlier (Faria and Bagajewicz, 2008).
- *Convex Envelopes* (McCormick, 1976) as used by Karuppiah and Grossmann (2006a).

To deal with the product of continuous variables and binary variables, we consider three variants of each procedure.

We now explain the basics of these techniques using the following generic case. Consider z to be the product of two continuous variables x and y:

$$z = x\, y \tag{1}$$

where both x and y subject to certain bounds:

$$x^L \le x \le x^U \tag{2}$$

$$y^L \le y \le y^U \tag{3}$$

Assume now that variable *y* is discretized using *D-1* intervals. The starting point of each interval is given by.

$$\hat{y}_d = y^L + (d-1)\frac{(y^U - y^L)}{D-1} \quad \forall d = 1..D \qquad y^L \le y \le y^U \tag{4}$$

In the case of the *direct discretization,* we simply substitute the variable *y* by its discrete values and allow the bilinear term (z) to be inside of one of the intervals, that is, between two successive discrete values. Binary variables (v_d) are used to assure that only one interval is picked.

$$\sum_{d=1}^{D-1} \hat{y}_d \, v_d \le y \le \sum_{d=1}^{D-1} \hat{y}_{d+1} \, v_d \tag{5}$$

$$\sum_{d=1}^{D-1} v_d = 1 \tag{6}$$

$$z \le x \sum_{d=1}^{D-1} \hat{y}_{d+1} \; v_d \tag{7}$$

$$z \ge x \sum_{d=1}^{D-1} \hat{y}_d \; v_d \tag{8}$$

Equation (5) states that y falls within the interval corresponding to the binary variable v_d, of which only one is equal to one (Equation (6) enforces this). This is done for the discretization variables, but if x (or a subset of it) is the BC variable, then a similar discretization as the one in (5) and (6) is included.

In turn, equations (6) and (7) bound the value of z to correspond to a value of y in the given interval.

In the case of using *McCormick's envelopes* for each interval, the equations are:

$$z \ge x^L y + \sum_{d=1}^{D-1} \left(x \, \hat{y}_d \, v_d - x^L \hat{y}_d \, v_d \right) \tag{9}$$

$$z \ge x^U y + \sum_{d=1}^{D-1} \left(x \, \hat{y}_{d+1} \, v_d - x^U \hat{y}_{d+1} \, v_d \right) \tag{10}$$

$$z \le x^L y + \sum_{d=1}^{D-1} \left(x \, \hat{y}_{d+1} \, v_d - x^L \hat{y}_{d+1} v_d \right) \tag{11}$$

$$z \le x^U y + \sum_{d=1}^{D-1} \left(x \, \hat{y}_d \, v_d - x^U \hat{y}_d v_d \right) \tag{12}$$

which are used in conjunction with equations (5) and (6).

When x (or a subset of it) is the BC variable, then we only add equations (5) and (6) for these variables, but do not incorporate the bounds of each interval in the above equations (9) through (12).

Note that even if the bilinearity generated by the multiplication of y and x was eliminated, we still have variable x being multiplied by the binary variable v_d in both cases. Once again there are different ways to linearize the product of a continuous and binary variable. These methods, in various forms, are very well known and we present next our implementation.

3.1.1. Direct Discretization Variants

When using the *direct discretization*, we linearize the product of x and the binary variable v_d using three different procedures:

- *Direct Discretization Procedure* 1, (DDP1); Let w_d be a positive variable ($w_d \geq 0$), such that $w_d = x\, v_d$. Then (7) and (8) are substituted by:

$$z \leq \sum_{d=1}^{D-1} \hat{y}_{d+1}\ w_d \tag{13}$$

$$z \geq \sum_{d=1}^{D-1} \hat{y}_d\ w_d \tag{14}$$

  and w_d is now obtained from the following linear equations:

$$w_d - x^U v_d \leq 0 \tag{15}$$

$$(x - w_d) - x^U (1 - v_d) \leq 0 \tag{16}$$

$$x - w_d \geq 0 \tag{17}$$

  Indeed, if v_d=0, equation (15) together with the fact that $w_d \geq 0$ renders, $w_d = 0$. Conversely, if v_d=1, equations (16) and (17) render $w_d = x$, which is what is desired. There is, however, an alternative more compact way of writing the linearization: Indeed, the following equations accomplish the same linearization.

- *Direct Discretization Procedure* 2 (DDP2): In this case, the product of the binary variable and the continuous variable is linearized as follows:

$$w_d \leq x^U v_d \qquad \forall d = 1..D-1 \tag{18}$$

$$w_d \geq x^L v_d \qquad \forall d = 1..D-1 \tag{19}$$

$$x = \sum_{d=1}^{D-1} w_d \tag{20}$$

Equations (18) and (19) guarantee that only one value of w_d (when v_d=1) can be greater than zero and in between bounds (all other w_d, for when v_d=0, are zero). Thus, equation (20) sets w_d to the value of *x*.

- *Direct Discretization Procedure* 3 (DDP3): This procedure uses the following equations to linearize equations (7) and (8):

$$z \le x\,\hat{y}_{d+1} + x^U\left(y^U - \hat{y}_{d+1}\right)\left(1 - v_d\right) \qquad \forall d = 1..D-1 \tag{21}$$

$$z \ge x\,\hat{y}_d - x^U\,\hat{y}_d\left(1 - v_d\right) \qquad \forall d = 1..D-1 \tag{22}$$

$$z \le x^U y \tag{23}$$

Equations (21) and (22) force z to be inside a chosen interval d^* (the one for which v_{d^*}=1). Indeed, when v_{d^*}=1, (21) and (22) reduces to the following inequalities: $x\,\hat{y}_{d^*} \le z \le x\,\hat{y}_{d^*+1}$. In turn, equations (5) and (23) reduce to $z \le x^U y^* \le x^U\,\hat{y}_{d^*+1}$ (we use y^* to denote the optimal value of y). In the other intervals where v_d=0, equations (22) and (23) reduce to $\left(x - x^U\right)\hat{y}_d \le z \le x^U y^* \le x^U\,\hat{y}_{d^*+1}$, which puts *z* between a lower negative bound and the right upper bound. Finally, equation (21) reduces to $z \le x\,\hat{y}_{d+1} + x^U(y^U - \hat{y}_{d+1})$, which is a valid inequality. We now need to show that equation (22) is also satisfied. For this, we recall that $x\,\hat{y}_{d^*} \le z \le x\,\hat{y}_{d^*+1}$. Then, for $d \ge d^*$ we have $\hat{y}_{d^*+1} \le \hat{y}_{d+1}$ and then, $z \le x\,\hat{y}_{d+1} + x^U(y^U - \hat{y}_{d+1}) \le x\hat{y}^*_{d+1} + x^U(y^U - \hat{y}_{d+1})$, which is a valid upper bound for that *d*. Conversely, when *d*<*d**, we have $\hat{y}_{d^*+1} > \hat{y}_{d+1}$ and then, $z \le x\,\hat{y}_{d+1} + x^U(y^U - \hat{y}_{d+1}) \le x\hat{y}_{d+1} + x^U(y^U - \hat{y}^*_{d+1})$. Adding and subtracting $x\hat{y}^*_{d+1}$ to the last term and rearranging, we get $z \le x\hat{y}^*_{d+1} + x^U(y^U - \hat{y}^*_{d+1}) - x(\hat{y}^*_{d+1} - \hat{y}_{d+1})$. Finally, noticing that $\hat{y}_{d+1} \le y^U$, one can write $z \le x\hat{y}^*_{d+1} + (x^U + x)(y^U - \hat{y}^*_{d+1})$

With all these substitution any MINLP model containing a bilinearity is transformed into an MILP, which is a lower bound of the original problem; this is because of the relaxation introduced.

3.1.2. McCormick Envelopes Variants

In this case, equations (9) through (12) are substituted by the following equations:

$$z \geq x^L\, y + \sum_{d=1}^{D-1} \left(\hat{y}_d\, w_d - x^L\, \hat{y}_d v_d \right) \tag{24}$$

$$z \geq x^U\, y + \sum_{d=1}^{D-1} \left(\hat{y}_{d+1}\, w_d - x^U\, \hat{y}_{d+1} v_d \right) \tag{25}$$

$$z \leq x^L\, y + \sum_{d=1}^{D-1} \left(\hat{y}_{d+1}\, w_d - x^L\, \hat{y}_{d+1} v_d \right) \tag{26}$$

$$z \leq x^U\, y + \sum_{d=1}^{D-1} \left(\hat{y}_d\, w_d - x^U \hat{y}_d v_d \right) \tag{27}$$

and several variants of how to linearize $w_d = x\, v_d$ follow:

- *McCormick's Envelopes Procedure* 1 (MCP1): It is when equations (15) to (17) are used.
- *McCormick's Envelopes Procedure* 2 (MCP2): In this case equations (18) to (20) are used instead of equations (15) to (17).
- *McCormick's Envelopes Procedure* 3 (MCP3): In this case, equations (5) and (6) are still used, but equations (9) to (12) are substituted by:

$$z \geq x^L\, y + x\ \ \hat{y}_d - x^L\, \hat{y}_d v_d - \left(x^L y^U + x^U \hat{y}_d \right)\left(1 - v_d\right) \qquad \forall d = 1..D-1 \tag{28}$$

$$z \geq x^U\ y + x\ \ \hat{y}_{d+1} - x^U\ \hat{y}_{d+1} v_d - x^U \left(y^U + \hat{y}_{d+1} \right)\left(1 - v_d\right) \qquad \forall d = 1..D-1 \tag{29}$$

$$z \leq x^L\, y + x\ \ \hat{y}_{d+1} - x^L\, \hat{y}_{d+1} v_d + \left(x^U y^U - x^L \left(y^L + \hat{y}_{d+1} \right)\right)\left(1 - v_d\right) \qquad \forall d = 1..D-1 \tag{30}$$

$$z \leq x^U\, y + x\ \ \hat{y}_d - x^U\, \hat{y}_d v_d + \left(x^U \left(y^U - y^L \right) - x^L\, \hat{y}_d \right)\left(1 - v_d\right) \qquad \forall d = 1..D-1 \tag{31}$$

$$z \leq x^U y \tag{32}$$

The case $x^L = 0$ is a very common situation in flowsheet superstructure optimization where connections between units exist formally but a flowrate of zero through some of these connections is almost always part of the optimal solution. If *x* represents the flowrates and *y* the composition of the stream, when $x^L = 0$ equation (28) reduces to $z \geq x\ \ \hat{y}_d - x^U \hat{y}_d \left(1 - v_d\right)$, which is the same as equation (22). In turn equation (30) reduces to $z \leq x\ \ \hat{y}_{d+1} + x^U y^U \left(1 - v_d\right)$, which does not reduce to (21).

As in the case of the direct discretization, when these equations are substituted in the original MINLP, they transform it into an MILP, which is a lower bound of the original problem.

Although one would expect the lower bound to be higher when the number of intervals increases, we have observed occasional drops.

In addition, as we shall see in the examples, which variables should be discretized in a bilinear term is also not straightforward. In many cases, the number of binary variables is much higher for one variable, but the solution could be found faster. This is the case of problems with component balances: flowrates participate in all the balances, whereas each balance contains its own composition. Conversely, discretizing flowrates may render a smaller number of integers but may affect speed of convergence. This is discussed in more detail below when we illustrate the method.

3.1.3. Discretization of both bilinear variables

Discretizing both variables has some advantages. First, the LB may improve in some schemes. If bound contraction on concentrations is successful, then further bound contraction of flowrates may take place. We reproduce the exact equations we considered for clarity and completeness in the appendix. In the few cases we tried, we did not observed improvements using the discretization of both variables mainly because the computation time is increased and the LB was observed to be at least as tight as discretizing concentrations.

3.2 *Concave functions*

Univariate functions used to estimate capital cost are often concave and expressed as functions of equipment sizes as follows:

$$z = \Omega\, y^{\alpha} \tag{33}$$

where α is often a value between 0 and 1, and y is the equipment capacity.

We first consider that variable y is discretized in several intervals as shown in equation (4). Then we linearize this concave function in each interval following Karuppiah and Grossmann (2006) as follows:

$$y^{\alpha} = \overline{y} \geq \sum_{d=1}^{D-1} v_d \left(\left(\hat{y}_d\right)^{\alpha} + \left(\frac{\left(\hat{y}_{d+1}\right)^{\alpha} - \left(\hat{y}_d\right)^{\alpha}}{\hat{y}_{d+1} - \hat{y}_d} \right) \left(y - \hat{y}_d \right) \right) \tag{34}$$

$$z = \Omega\, \overline{y} \tag{35}$$

which we use in conjunction with (5) and (6).

Note that, again, we have the product of a binary variable (v_d) and a continuous variable (y). The linearization of equation (34) is the following:

$$\overline{y} \geq \sum_{d=1}^{D-1} \left(\left(\hat{y}_d\right)^{\alpha} v_d + \left(\frac{\left(\hat{y}_{d+1}\right)^{\alpha} - \left(\hat{y}_d\right)^{\alpha}}{\hat{y}_{d+1} - \hat{y}_d} \right) \left(\beta_d - \hat{y}_d v_d\right) \right) \tag{36}$$

$$y = \sum_{d=1}^{D-1} \beta_d \tag{37}$$

$$\beta_d \leq \hat{y}_{d+1} v_d \qquad \forall d = 1..D-1 \tag{38}$$

$$\beta_d \geq \hat{y}_d v_d \qquad \forall d = 1..D-1 \tag{39}$$

which is again used in conjunction with (5) and (6). When substituted in the original MINLP, they transform it into an MILP. Such MILP is a lower bound of the original problem if z only appears in the objective function as an additive term, together with the equation defining it (equation 33). Conversely, when z shows up in some constraint of the problem, but not in the objective as an additive term, then one would have to add an overestimator like the following:

$$\overline{y} \leq \sum_{d=1}^{D-1} v_d \left(\left(\frac{\hat{y}_d + \hat{y}_{d+1}}{2} \right)^{\alpha} + \alpha \left(\frac{\hat{y}_d + \hat{y}_{d+1}}{2} \right)^{\alpha-1} \left(y - \frac{\hat{y}_d + \hat{y}_{d+1}}{2} \right) \right) \tag{40}$$

which uses the tangent line at the middle of the interval as an upper bound.

4. Bound Contraction -Interval Elimination Strategy

Once a problem has been linearized and solved, the solution from this LB is used to obtain good guesses for solving the upper bound problem (the original problem is used in most cases). Once a lower bound and an upper bound have been found there is a need to identify which intervals can be eliminated from consideration. The lower bound solution points at a set of intervals, one per variable. This solution is used to find an upper bound and also to guide the elimination of certain intervals. The procedure is as follows:

- Step 1. Run the lower bound model with no forbidden intervals and re-discretized variables over the range that survived.
- Step 2. Use the solution from the lower bound as an initial point to solve the full NLP or MINLP problem to obtain an Upper Bound.
- Step 3. If the gap between the upper bound and the lower bound is lower than the tolerance, the solution was found. Otherwise go to Step 4.
- Step 4. Run the lower bound model, this time forbidding the interval that contains the answer for the first discretized variable.
- Step 5. If the new problem is infeasible, or if feasible and the objective function is higher than the current upper bound, then all the intervals of this variable, except the original one that was forbidden, are eliminated. The surviving feasible region between the new bounds is discretized again.
- Step 6. Repeat the procedure for all the other variables, one at a time.
- Step 7. Go back to Step 1.

Note that to guarantee the optimality, not all of the lower bound models need to be solved to zero gap. The only problems that need to have zero gap are the ones in which the lower bound of the problem (or sub-problems) are obtained, which is done in step 1. The lower bound models used to eliminate intervals (step 4) can be solved to feasibility between its lower bound and the current upper bound, which is always set as the upper bound of the whole procedure.

In some cases, a pre-processing step using bound arithmetic to reduce the initial bounds of certain variables can be performed. We discuss the specifics of this in part II. .

The above is the standard version of our interval elimination (bound contraction) procedure, which we call *One-pass with one forbidden interval elimination* because the elimination process takes place sequentially, only one variable at a time and only once for each variable.

Variations to the above elimination strategy are possible. We distinguish five specific set of alternative options

- Options related to the amount of times all variables are considered for bound contraction:

    - *One-Pass Elimination:* In Step 6, each variable is visited only once before a new lower bound of the whole problem is obtained.
    - *Cyclic Elimination:* In Step 6, once all variables are visited, the method returns to the first variable and starts the process again, as many times as needed, until no more bound contraction is achieved.

- Options related to the amount of times each variable is bound contracted:

    - *Exhaustive elimination:* In Step 6, once each variable is contracted, the process is repeated again for that same variable until no bound contraction takes place. Only then, the process moves to the next variable.
      each variable is visited only once before a new lower
    - *Non-Exhaustive elimination:* In Step 6, once each variable is contracted once, the process moves to the next variable.

- Options related to the updating of the UB

    - *Active Upper Bounding:* Each time an elimination takes place, the upper bound is calculated again. This helps when the gap between lower and upper bound (feasible solution) improves too slowly.
    - *Active Lower Bounding*: Each time an elimination takes place, the lower bound solution calculated again. In such case, one would allow all surviving intervals, and rediscretize them. If the gap between LB and UB is within tolerance one can terminate the entire procedure. This option could be really attractive if several variables are used.

- Options related to the amount of intervals used for forbidding

    - *Single interval forbidding*: This consists of forbidding only the interval that brackets the solution
    - *Extended interval forbidding*: This consists of forbidding the interval identified originally plus some number of adjacent ones. This is efficient when a large number of intervals are used to obtain lower bounds. Adjacent intervals, if left not forbidden, may render lower bounds that are not larger than the current upper bound. Thus, by forbidding them, other intervals are forced to be picked and those may render larger LB and lead to elimination.

- Options related to the amount of variables that are forbidden

    - *Single Variable Elimination*: This procedure is the one outlined above.
    - *Collective Elimination*: This procedure consists of forbidding the combination of the intervals identified in the lower bound. We anticipate having problems with this strategy when the size of the problem is large

When no interval is eliminated and the lower bound-upper bound gap is still larger than the tolerance, one can resort to increase the number of intervals and start over. This procedure normally renders better lower bounds and more efficient eliminations when the *Extended interval forbidding is applied*. When our standard option, the *One-pass with one forbidden interval elimination,* is used, an increase in the number of intervals will select a smaller part of the feasible range of each variable. Thus, increasing the number of intervals helps because it provides tighter lower bounds. However, a large number of intervals can also significantly increase the running time.

5. Branch and Bound Procedure

It is possible that the above interval elimination procedure fails to reduce the gap that is even using the maximum number of intervals, no interval eliminations are possible. In such a situation, we resort to a branch bound procedure. In many methods addressing bilinear terms directly (Adhyla et al., 1999; Karuppiah and Grossmann, 2006a), the branching is normally done in the variable that is being discretized. However one can branch on the other or on both. In our case, we branch on the continuous variables by splitting their interval from lower to upper bound in two intervals. As the non-discretized variables participate on the lower bound models and thus influence their tightness, the generation of sub-problems with different non-discretized variables bounds can speed up the procedure.

For branching we use one of the following two criteria:

- The new continuous variable that is split in two is the one that has the largest deviation between the value of $z_{i,j}^{LB}$ in the parent node and the product of the corresponding variables x_i^{LB} and y_j^{LB}, that is choose the variable i that satisfies the following

$$Max_i \left\{ \left| z_{i,j}^{LB} - x_i^{LB} \; y_j^{LB} \right| \right\} \tag{41}$$

- Using information of the current upper bound solution: We do this by choosing the variable that contributes to the largest gap between z's from the lower and upper bound, that is, we choose the variable i that satisfies the following

$$Max_i \left\{ \left| z_{i,j}^{LB} - z_{i,j}^{UB} \right| \right\} \tag{42}$$

In addition to the B&B procedure, at each node we perform as many interval eliminations (bound contractions) as possible.

6. Similarities and Differences with other Methods

Our methodology borrows and intersects several other previously presented discretization and underestimation methods that render lower bounds. For example, we are considering the use of direct discretization instead of McCormick (which is supposedly tighter). The advantages would be to verify if it runs faster and consequently is able to find the solution faster (even if using more iterations).

7. Implementation issues

Our methodology requires making several choices. These choices are:

- *Variables to be discretized*. In water management and pooling problems these could be concentrations, flowrates, or both.
- *Number of discrete intervals per variable*: It does not need to be the same for all variables.
- *LB model*: DDP1, DDP2, DDP3, MCP1, MCP2, or MCP3.
- *Variables chosen to perform Bound Contraction*: They need not be the same as the once chosen to be discretized. For example, one can discretize concentrations and build a DDP1-LB model based on these discretization, but perform bound contraction on flowrates. For this, one needs to discretize the flowrates as well. The LB-Model, however, would not consider other than continuous flowrates, only including equation (5) for flowrates to bracket the flowrate value and to be able to forbid it.
- *Elimination strategy*: The standard One-pass with one forbidden interval elimination, or the variants (One pass or Cyclic Elimination, Exhaustive or

not Exhaustive Elimination, Active Upper/Lower Bounding or not, Single vs. Extended Intervals forbidding, or Collective elimination,).

- *Variables to partition in the Branch and bound procedure*.

With such a large amount of options, it is cumbersome to explore all of them. In our examples, when we report successful cases (i.e., those that seem to work fast and better than other procedures), we made no attempt to explore the possibility of other combinations being even more efficient computationally. We also make an effort to show some variant's success, even though they are less efficient. For the examples for which the method is not as quick and efficient we report the best we could achieve and mention a few of the failures.

8. Illustration of the Interval Elimination procedure

8.1 Elimination procedure

We illustrate the details of the *One-pass with one forbidden interval elimination* procedure using a simple water network example from Wang and Smith (1994). This example optimizes only the water-using subsystem, which targets minimum freshwater consumption and has two water-using units and two contaminants.

The non-linear model used to describe this problem can be given by the following set of equations:

Water balance through the water-using units

$$\sum_{w} FWU_{w,u} + \sum_{u^* \neq u} FUU_{u^*,u} = \sum_{s} FUS_{u,s} + \sum_{u^* \neq u} FUU_{u,u^*} \qquad \forall u \tag{43}$$

where $FWU_{w,u}$ is the flowrate from freshwater source *w* to the unit *u*, $FUU_{u^*,u}$ is the flowrates between units *u** and *u*, $FUS_{u,s}$ is the flowrate from unit *u* to sink *s*.

Contaminant balance at the water-using units

$$\sum_{w} \left(CW_{w,c} \; FWU_{w,u} \right) + \sum_{u^* \neq u} ZUU_{u^*,u,c} + \Delta M_{u,c} = \sum_{u^* \neq u} ZUU_{u,u^*,c} + \sum_{s} ZUS_{u,s,c} \qquad \forall u,c \tag{44}$$

where $CW_{w,c}$ is concentration of contaminant *c* in the freshwater source *w*, $\Delta M_{u,c}$ is the mass load of contaminant *c* extracted in unit *u*, $ZUU_{u^*,u,c}$ is the mass flow of contaminant *c* in the stream leaving unit *u** and going to unit *u*, and $ZUS_{u,s,c}$ is the mass flow of contaminant *c* in the stream leaving unit *u* and going to sink s.

Maximum inlet concentration at the water-using units

$$\sum_{w}\left(CW_{w,c}\ FWU_{w,u}\right)+\sum_{u^*\neq u} ZUU_{u^*,u,c} \leq C_{u,c}^{in,\,max}\left(\sum_{w} FUW_{w,u}+\sum_{u^*\neq u} FUU_{u^*,u}\right) \qquad \forall u,c \tag{45}$$

where $C_{u,c}^{in,\,max}$ is the maximum allowed concentration of contaminant c at the inlet of unit u.

Maximum outlet concentration at the water-using units

$$\begin{aligned}&\sum_{w}\left(CW_{w,c}\ FWU_{w,u}\right)+\sum_{u^*} ZUU_{u^*,u,c}+\Delta M_{u,c}\\ &\qquad \leq C_{u,c}^{out,\,max}\left(\sum_{u^*} FUU_{u,u^*}+\sum_{r} FUR_{u,r}+\sum_{u^*} FUU_{u,u^*}\right) \quad \forall u,c\end{aligned} \tag{46}$$

where $C_{u,c}^{out,\,max}$ is the maximum allowed concentration of contaminant c at the outlet of unit u.

The contaminant mass flowrates are enough and no new variables are needed in the case of mixing nodes. However, the splitting nodes at the outlet of each unit and the regeneration units need constraints that will reflect that all these contaminant flows and the total mass flows are consistent with the concentrations of the different contaminants. Thus we add the corresponding relations:

Contaminant mass flowrates

$$ZUU_{u,u^*,c} = FUU_{u,u^*}\ C_{u,c}^{out} \quad \forall u,u^*,c \tag{47}$$

$$ZUS_{u,s,c} = FUS_{u,s}\ C_{u,c}^{out} \quad \forall u,s,c \tag{48}$$

where $C_{u,c}^{out}$ is the outlet concentration of contaminant c in unit u. Finally, we write:

$$C_{u,c}^{out,Min} \leq C_{u,c}^{out} \leq C_{u,c}^{out,Max} \qquad \forall u,s,c \tag{49}$$

Objective functions (Minimum freshwater consumption)

$$Min\ FW = \sum_{w}\sum_{u} FWU_{w,u} \tag{50}$$

Table 1 presents the limiting data of this problem.

Table 1 – Limiting data of example 1.

Process	Contaminant	Mass Load (kg/h)	$C^{in,max}$ (ppm)	$C^{out,max}$ (ppm)
1	A	4	0	100
	B	2	25	75
2	A	5.6	80	240

	B	2.1	30	90

For the illustration of this example, the pure discrete concentration lower bound is used with two initial intervals (Figure 1) and the elimination procedure is applied on the outlet concentrations of the water-using units. The standard strategy (one-pass non-exhaustive elimination) is used.

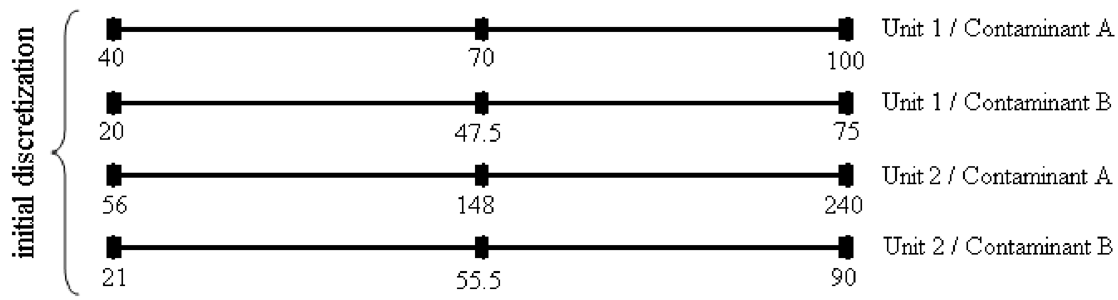

Figure 1 – Illustrative example of the discrete approach - initialization.

In the upper part of Figure 2 the results of the lower bound using the pre-processed bounds, which corresponds to a value of 52.89 t/h are depicted. Using the results from this lower bound as initial points, the full problem was run and the solution obtained (54 t/h) corresponds to the first upper bound of the problem.

When the lower bound model is re-run forbidding the interval corresponding to Unit 1/Contaminant A, that is, the interval 70 to 100 ppm is forbidden, the interval from 40 to 70 ppm is eliminated because forcing the lower bound in this interval renders a value of the LB higher than 54 t/h. The remaining part (70ppm to 100ppm) is rediscretized in two new intervals. Then the lower bound model is run forbidding the interval corresponding to Unit 1/Contaminant B, that is the interval 47.5 to 75 ppm. The solution is again higher than 54 t/h. Thus, the interval between 20ppm and 47.5 ppm is eliminated and the remaining is rediscretized. Applying this procedure to the rest of the variables renders eliminating the intervals shown in Figure 2.

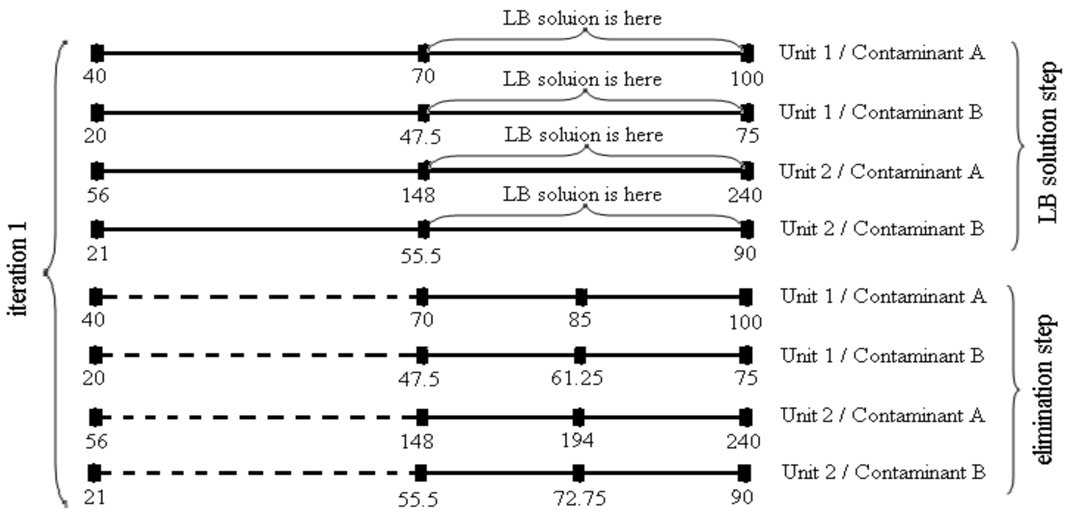

Figure 2 – Illustrative example of the discrete approach – 1st iteration.

After the first iteration the lower and upper bound do not change (LB = 52.90 t/h and UB = 54 t/h). The second iteration of the illustrative example is shown in Figure 3. The elimination procedure is repeated again, one variable at a time, and in all cases, the solutions found are larger than the current upper bound. Therefore, each time the corresponding interval in each variable is eliminated, the selected interval is re-discretized and the procedure moves to the next variable.

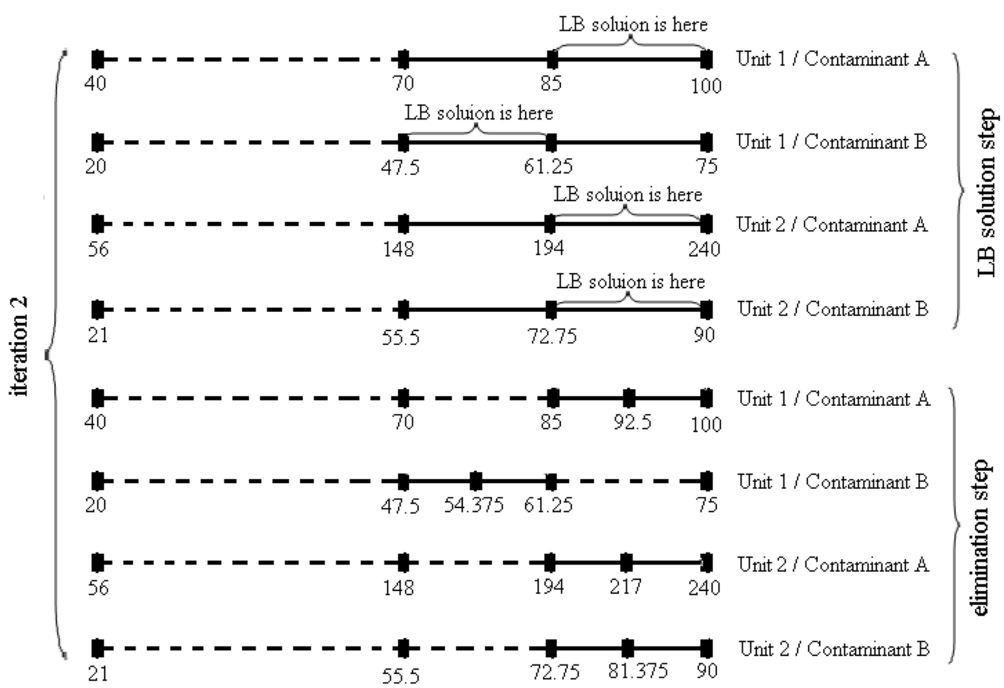

Figure 3 – Illustrative example of the discrete approach – 2nd iteration.

This procedure is repeated until the lower bound solution is equal (or has a given tolerance difference) to the upper bound solution. This illustrative example, using the DDP3 and discretizing concentrations in two intervals, is solved in 3 iterations and 0.60 seconds using a relative tolerance of 1%. The actual solution reaches 0.65% gap. Throughout this paper, we report solving times, not including pre-processing/compilation times.

Table 2 presents the progress of the solution through the iterations. The upper bound (54 t/h) is identified in the first iteration and is the global solution. The lower bound solution, however, does not improve until the third iteration. The optimum network of this example is presented in Figure 4.

The other option for the elimination step is cyclic non-exhaustive elimination. Table 3 and 4 show the progress of the solution when the cyclic non-exhaustive elimination is applied.

Table 2 – Solution progress of the illustrative example.

Iteration	Lower Bound	Upper Bound	Relative error	Intervals eliminated
0	52.90 t/h	54.00 t/h	2.02%	NA
1	52.90 t/h	54.00 t/h	2.02%	4
2	52.90 t/h	54.00 t/h	2.02%	4
3	53.65 t/h	54.00 t/h	0.65%	4

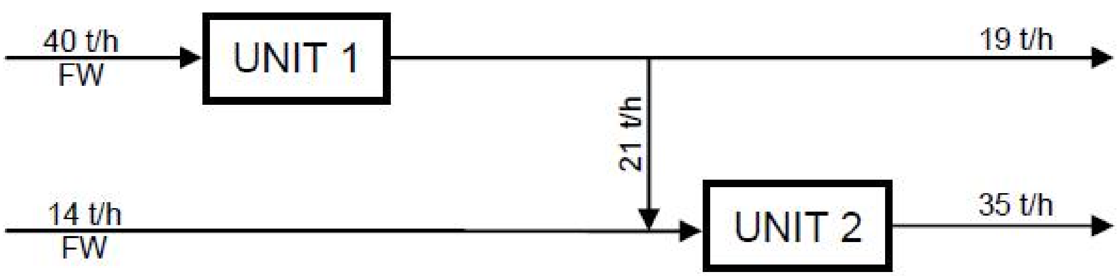

Figure 4 – Optimum network of example 1.

Table 3 – Solution progress of the illustrative example – using cyclic non-exhaustive elimination.

Iteration	Lower Bound	Upper Bound	Relative error	Number of cycles	Eliminations
0	52.90 t/h	54.00 t/h	2.02%	NA	NA
1	52.90 t/h	54.00 t/h	2.02%	4	10
2	53.67 t/h	54.00 t/h	0.62%	5	8

Table 4 – Number of elimination in each cycle – using cyclic non-exhaustive elimination.

Iteration	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5
1	4	3	2	1	NA
2	1	3	2	1	1

Despite the fact that this procedure takes a smaller number of iterations, the overall running time for this example was higher (2.26 s against 0.60 s using the one-pass elimination). This is expected because this is a small problem, in which the lower bounding (step 2) is not computationally expensive. Thus, unnecessary elimination (more than the needed to achieve the given tolerance gap) may occur if the lower bound is not often verified.

The solution using one-pass exhaustive elimination is also investigated. Table 5 shows the progress of the iterations and Table 6 shows which variable had its bounds contracted and how many eliminations existed in each iteration. This strategy took 1.30 seconds.

Table 5 – Solution progress of the illustrative example – using one-pass exhaustive elimination.

Iteration	Lower Bound	Upper Bound	Relative error	Eliminations
0	52.89 t/h	54.00 t/h	2.02%	NA
1	52.89 t/h	54.00 t/h	2.02%	10
2	53.67 t/h	54.00 t/h	0.62%	6

Table 6 – Exhaustive eliminations progress of the illustrative example – using one-pass exhaustive elimination.

Iteration	Cout			
	Unit 1 Contaminant A	Unit 1 Contaminant B	Unit 2 Contaminant A	Unit 2 Contaminant B
1	4	2	1	3
2	-	1	3	2

8.1 Effect of the Number of Intervals

The number of initial intervals has also influence on the performance of the proposed methodology. Since it is known that a continuous variable can be substituted by discrete values when the number of discrete values goes to infinity, it is expected that less iterations are needed when more discrete intervals are added. On the other hand, this generates a higher number of integer variables (what means a larger MILP model), and might make the problem computationally very expensive (increase the overall time to run it).

This influence is analyzed only for the cases of one-pass non-exhaustive elimination, which have presented the best option when only 2 intervals are considered. Additionally, the influence of the *Extended interval forbidding* option is also verified. This option represents two main advantages: reduce the number of binary in the elimination step; and, facilitate eliminations. On the other hand, when only one interval is forbidden and an elimination takes place, the discharged portion of the variable is lager then if the *Extended interval forbidding* option was used and the stopping criteria is when the tolerance is satisfied. We performed an analysis for the above example and the results are shown in Figures 5 and 6, where the number of intervals is increased to twenty two.

For the *One-pass with one forbidden interval elimination* option, the quickest solution (0.07s) is found when the procedure is initialized with 7 intervals. This is the point in which the solution is first found at the root node. For the *Extended interval forbidding* case, very similar CPU times are found for the cases in which the solution is found at the root node (7, 9, 11 and 13 to 18 intervals), that is, computational times of approximately 0.15 s.

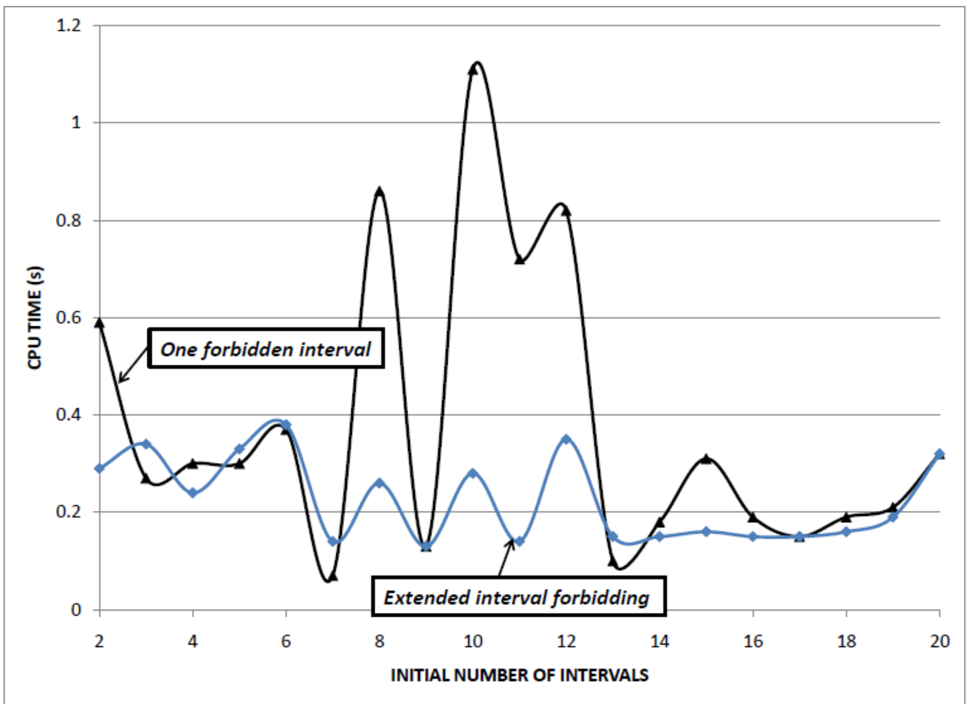

Figure 5 – Influence of the number of initial intervals and the use of *Extended interval forbidding* option – CPU time.

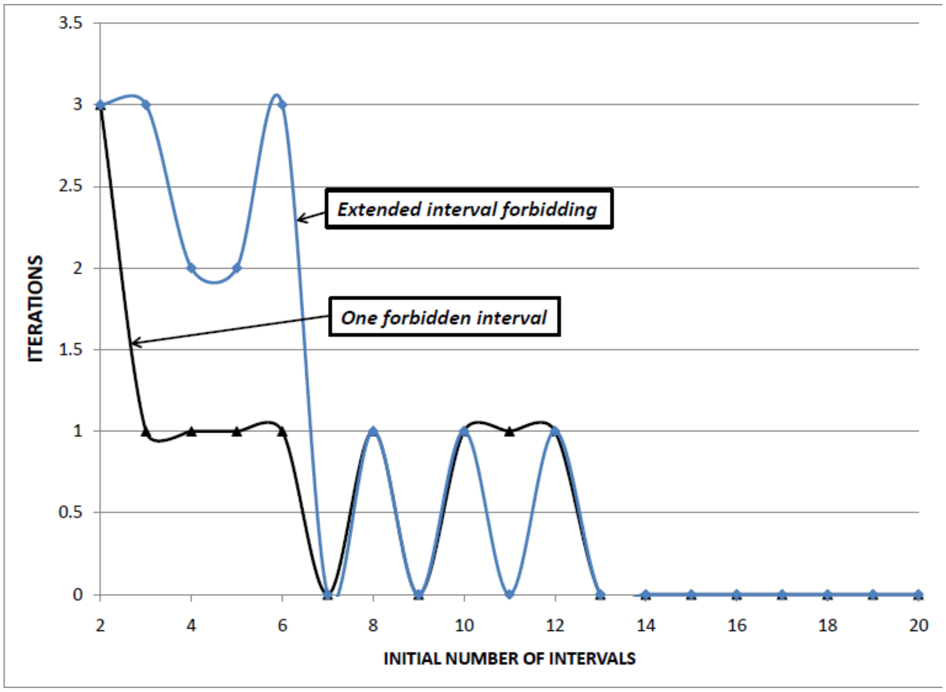

Figure 5 – Influence of the number of initial intervals and the use of *Extended interval forbidding* option - Iterations.

9. *Conclusions*

We presented the theory of a new bound contraction method that uses discrete portions of the feasible region to identify if they can part of the global solution or not. This procedure is used to contract the bounds of one (or both) variables participating in bilinearities. After the contraction further discretization leads to a tighter lower bound model. In addition to the above, the lower bound model is constructed discretizing at least one of the variables participating in all bilinear terms. Variables participating in bound contraction may be these variables or others. When bound contraction is exhausted, the procedure can resorts to one of the following alternatives: increased discretization of the lower bound model, or, creating sub-problems using a branch and bound scheme. The sub-problems are created by partitioning the interval between bounds of variables that need not be exactly the same as the variables used for discretization or bound contraction. Picking at least one of the variables used for bound contraction or for discretization in the lower bound model. In the second part of this paper we present results.

Acknowledgement

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

Nomenclature

Sets:

u - Water using units
w - Freshwater sources
s - Wastewater sinks
c - Contaminants
d - Concentration discrete points

Parameters:

fwc_w - Freshwater cost

wwc_s - Wastewater treatment cost

$Fmax$ - Maximum flowrate allowed

$Fmin$ - Minimum flowrate allowed

CFW_w - Cost of freshwater

CWW_s - Cost of wastewater

$CW_{w,c}$ - Freshwater concentrations

$Cinmax_{u,c}$ - Maximum inlet concentrations at each water-using unit

$Coutmax_{u,c}$ - Maximum outlet concentrations at each water-using unit

$CSmax_{s,c}$ - Maximum concentrations at the sinks

$\Delta M_{u,c}$ - Mass load of each contaminant at each water-using unit

$LWU_{w,u}$ - Distance between a freshwater source and a water-using unit

$LUU_{u,u*}$ - Distance between a freshwater two water-using units

$LUS_{u,s}$ - Distance between a water-using unit and a wastewater sink

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)

df - Discount factor

$DC_{d,c,u}$ - Discrete concentration of each contaminant in each unit

Variables:

$FW_{w,u}$ - Flowrate between a freshwater source and a water-using unit

$F_{u,u*}$ - Flowrate between two water-using units

$FS_{u,s}$ - Flowrate between a water-using unit and a wastewater sink

$ZU_{u,u*,c}$ - Mass load of each contaminant in the stream between two water-using units

$ZS_{u,s,c}$ - Mass load of each contaminant in the stream between a water-using unit and a wastewater sink

$ZM_{u,c}$ - Mass load of each contaminant at the mix point of a water-using unit

$Consu$ - Freshwater consumption

CU_u - Outlet concentration of each unit

<u>Binary Variables:</u>

YU_{u,u^*} - Existence of flowrate between two water-using units

$YS_{u,s}$ - Existence of flowrate between a water-using unit and a wastewater sink

$XU_{u,u^*,c,d}$ - Concentration interval choice of each contaminant in the stream between two water-using units

$XS_{u,s,c,d}$ - Concentration interval choice of each contaminant in the stream between a water-using unit and a wastewater sink

$XM_{u,c,d}$ - Concentration interval choice of each contaminant in mix point of a water-using unit

$XUout_{u,c,d}$ - Concentration interval choice of each contaminant at the outlet of a water-using unit

REFERENCES

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

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

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

Bergamini, M.L., Aguirre, P. and 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. and Bagajewicz, M.J. (2008). A new approach for the design of multicomponent water/wastewater networks. *Computers and Chemical Engineering, 25, 43-48*.

Floudas, C.A.(2000a), Deterministic Global Optimization: Theory,Methods andApplications. *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.: Global optimization in the 21st century: advances and challenges. Comput. Chem. Eng. **29**(6), 1185–1202 (2005)

Floudas, C.A. and Gounaris, C.E. (2009). A review of recent advances in global optimization. *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., and 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*.

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. and 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. and Leclerc, A. (1992). Rigorous Methods for Global Optimization. *Recent Advances in Global Optimizarion (Edited by C. A. Floudas and P. M. Pardalos), p. 321. Princeton University Press, Princeton*.

Padberg, M. (2000). Approximating separable nonlinear functions via mixed zero-one programs. *Operations research Letters, 27, 1-5*.

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

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

Ruiz, J.P. and Grossmann, I.E. (2009). Strengthening of Lower Bounds in the Global Optimization of Bilinear and Concave Generalized Disjunctive Programs. *Computer and Chemical Engineering (accepted).*

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

Sherali, H.D., Adams, W.P. (1999). A Reformulation-Linearization Technique for solving Discrete and Continuous Nonconvex Problems. Nonconvex Optimization and itsApplications. *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. And I. E. Grossmann (1999), A Branch and Contract Algorithm for Problems with Concave Univariate, Bilinear and Linear Fractional Terms. Journal of Global Optimization, 14: 217–249.

APPENDIX

In this appendix we write the equations for the discretization of both variables in a bilinear terms. For these we use discretization variables w_{d_y} for *x* and s_{d_y} for *y*. In addition we use binary variables v_{d_y} for *x* and r_{d_x} for *y*.

For all DDP1, DDP2 and DDP3, we have :

$$\sum_{d_y=1}^{D_y-1} \hat{y}_{d_y} v_{d_y} \leq y \leq \sum_{d_y=1}^{D_y-1} \hat{y}_{d_y+1} v_{d_y} \tag{A-51}$$

$$\sum_{d_y=1}^{D_y} v_{d_y} = 1 \tag{A-52}$$

$$z \leq \sum_{d_y=1}^{D_y-1} \hat{y}_{d_y+1} \; w_{d_y} \tag{A-53}$$

$$z \geq \sum_{d_y=1}^{D_y-1} \hat{y}_{d_y} \; w_{d_y} \tag{A-54}$$

$$\sum_{d_x=1}^{D_x-1} \hat{x}_{d_x} r_{d_x} \leq x \leq \sum_{d_x=1}^{D_x-1} \hat{x}_{d_x+1} r_{d_x} \tag{A-55}$$

$$\sum_{d_x=1}^{D_x} r_{d_x} = 1 \tag{A-56}$$

$$z \leq \sum_{d_x=1}^{D_x-1} \; s_{d_x} \; \hat{x}_{d_x+1} \tag{A-57}$$

$$z \leq \sum_{d_x=1}^{D_x-1} \; s_{d_x} \; \hat{x}_{d_x} \tag{A-58}$$

Additional equations for DDP1:

$$w_{d_y} - x^U v_{d_y} \leq 0 \tag{A-59}$$

$$(x - w_{d_y}) - x^U (1 - v_{d_y}) \leq 0 \tag{A-60}$$

$$x - w_{d_y} \geq 0 \tag{A-61}$$

$$r_{d_x} - y^U s_{d_x} \leq 0 \tag{A-62}$$

$$(y - s_{d_y}) - y^U (1 - r_{d_y}) \leq 0 \tag{A-63}$$

$$y - s_{d_y} \geq 0 \tag{A-64}$$

Additional equations for DDP2:

$$w_{d_y} \le x^U v_{d_y} \quad \forall d_y = 1..D_y - 1 \tag{A-65}$$

$$w_{d_y} \ge x^L v_{d_y} \quad \forall d_y = 1..D_y - 1 \tag{A-66}$$

$$x = \sum_{d_y=1}^{D_y-1} w_{d_y} \tag{A-67}$$

$$s_{d_x} \le y^U r_{d_x} \quad \forall d_x = 1..D_x - 1 \tag{A-68}$$

$$s_{d_x} \ge y^L r_{d_x} \quad \forall d_x = 1..D_x - 1 \tag{A-69}$$

$$y = \sum_{d_x=1}^{D_x-1} s_{d_x} \tag{A-70}$$

Additional equations for DDP3:

$$z \le x\,\hat{y}_{d_x+1} + x^U\left(y^U - \hat{y}_{d_x+1}\right)\left(1 - v_{d_x}\right) \qquad \forall d_x = 1..D_x - 1 \tag{A-71}$$

$$z \ge x\,\hat{y}_{d_x} - x^U \hat{y}_{d_x}\left(1 - v_{d_x}\right) \qquad \forall d_x = 1..D_x - 1 \tag{A-72}$$

$$z \le x^U y \tag{A-73}$$

$$z \le \hat{x}_{d_y+1} y + x^U\left(y^U - \hat{x}_{d_y+1}\right)\left(1 - r_{d_y+1}\right) \qquad \forall d_y = 1..D_y - 1 \tag{A-74}$$

$$z \ge \hat{x}_{d_y} y - x^U \hat{x}_{d_y}\left(1 - r_{d_y}\right) \qquad \forall d_y = 1..D_y - 1 \tag{A-75}$$

$$z \le y^U x \tag{A-76}$$

There is an alternative approach that relies on introducing a introduce a variable that will be one if both intervals are chosen. For example, DDP1 one would write:

$$z \le \sum_{d_x=1}^{D_x-1} \sum_{d_y=1}^{D_y-1} \hat{x}_{d_x+1}\ \hat{y}_{d_y+1}\ \zeta_{d_x,d_y} \tag{77}$$

$$z \ge \sum_{d_x=1}^{D_x-1} \sum_{d_y=1}^{D_y-1} \hat{x}_{d_x}\ \hat{y}_{d_y}\ \zeta_{d_x,d_y} \tag{78}$$

$$\sum_{d_x,d_y} \zeta_{d_x,d_y} = 1 \qquad \forall d_x, d_y \tag{79}$$

$$\zeta_{d_x,d_y} \le r_{d_x} \qquad \forall d_x, d_y \tag{80}$$

$$\zeta_{d_x,d_y} \le v_{d_y} \qquad \forall d_x, d_y \tag{81}$$

where ζ_{d_x,d_y} can be continuous. Clearly this introduces an additional fairly large number of new variables, which we believe may not be the only disadvantage, as the lower bound is also less tight than the alternative (A-9 through A-14).

The McCormick double discretization schemes (MCP1 and MCP2) make use of the cross variable selecting variable ζ_{d_x,d_y}. For MCP1, we write:

$$z \geq \sum_{d_x=1}^{D-1}\left(\hat{x}_{d_x} s_{d_x} \right) + \sum_{d_y=1}^{D-1}\left(w_{d_y} \hat{y}_{d_y} \right) - \sum_{d_x=1}\sum_{d_y=1}\left(\hat{x}_{d_x} \hat{y}_{d_y} \zeta_{d_x,d_y} \right) \tag{A-82}$$

$$z \geq \sum_{d_x=1}^{D-1}\left(\hat{x}_{d_x+1} s_{d_x} \right) + \sum_{d_y=1}^{D-1}\left(w_{d_y} \hat{y}_{d_y+1} \right) - \sum_{d_x=1}^{D-1}\sum_{d_y=1}^{D-1}\left(\hat{x}_{d_x+1} \hat{y}_{d_y+1} \zeta_{d_x,d_y} \right) \tag{A-83}$$

$$z \leq \sum_{d_x=1}^{D-1}\left(\hat{x}_{d_x+1} s_{d_x} \right) + \sum_{d_y=1}^{D-1}\left(w_{d_y} \hat{y}_{d_y} \right) - \sum_{d_x=1}\sum_{d_y=1}\left(\hat{x}_{d_x+1} \hat{y}_{d_y} \zeta_{d_x,d_y} \right) \tag{A-84}$$

$$z \leq \sum_{d_x=1}^{D-1}\left(\hat{x}_{d_x} s_{d_x} \right) + \sum_{d_y=1}^{D-1}\left(w_{d_y} \hat{y}_{d_y+1} \right) - \sum_{d_x=1}\sum_{d_y=1}\left(\hat{x}_{d_x} \hat{y}_{d_y+1} \zeta_{d_x,d_y} \right) \tag{A-85}$$

$$\sum_{d_x,d_y} \zeta_{d_x,d_y} = 1 \qquad \forall d_x, d_y \tag{A-86}$$

$$\zeta_{d_x,d_y} \leq r_{d_x} \qquad \forall d_x, d_y \tag{A-87}$$

$$\zeta_{d_x,d_y} \leq v_{d_y} \qquad \forall d_x, d_y \tag{A-88}$$

$$w_{d_y} - x^U v_{d_y} \leq 0 \tag{A-89}$$

$$(x - w_{d_y}) - x^U (1 - v_{d_y}) \leq 0 \tag{A-90}$$

$$x - w_{d_x} \geq 0 \tag{A-91}$$

$$s_d - y^U r_{d_y} \leq 0 \tag{A-92}$$

$$(y - s_{d_y}) - y^U (1 - r_{d_y}) \leq 0 \tag{A-93}$$

$$y - s_{d_y} \geq 0 \tag{A-94}$$

For MCP2 we use equations (A-32) to (A-38) and replace (A-39) to (A-44) by:

$$w_{d_x} \leq x^U v_{d_x} \qquad \forall d_x = 1..D_x - 1 \tag{A-95}$$

$$w_{d_x} \geq x^L v_{d_x} \qquad \forall d_x = 1..D_x - 1 \tag{A-96}$$

$$x = \sum_{d_x=1}^{D-1} w_{d_x} \tag{A-97}$$

$$k_{d_y} \le y^U v_{d_y} \qquad \forall d_y = 1..D_y - 1 \tag{A-98}$$

$$k_{d_y} \ge y^L v_{d_y} \qquad \forall d_y = 1..D_y - 1 \tag{A-99}$$

$$y = \sum_{d_y=1}^{D-1} k_{d_y} \tag{A-100}$$

An alternative scheme without the cross variable ζ_{d_x,d_y} can be constructed as follows: For MCP1,

$$z \ge \sum_{d_x=1}^{D-1}\left(\hat{x}_{d_x} s_{d_x} \right) + \sum_{d_y=1}^{D-1}\left(w_{d_y} \hat{y}_{d_y} \right) - \sum_{d_y=1}\left(t_{d_y}\, \hat{y}_{d_y} \right) \tag{A-51}$$

$$z \ge \sum_{d_x=1}^{D-1}\left(\hat{x}_{d_x+1} s_{d_x} \right) + \sum_{d_y=1}^{D-1}\left(w_{d_y} \hat{y}_{d_y+1} \right) - \sum_{d_y=1}\left(t_{d_{y+1}}\, \hat{y}_{d_{y+1}} \right) \tag{A-52}$$

$$z \le \sum_{d_x=1}^{D-1}\left(\hat{x}_{d_x+1}\, s_{d_x} \right) + \sum_{d_y=1}^{D-1}\left(w_{d_y} \hat{y}_{d_y} \right) - \sum_{d_y=1}\left(q_{d_y}\, \hat{y}_{d_y} \right) \tag{A-53}$$

$$z \le \sum_{d_x=1}^{D-1}\left(\hat{x}_{d_x}\, s_{d_x} \right) + \sum_{d_y=1}^{D-1}\left(w_{d_y} \hat{y}_{d_y+1} \right) - \sum_{d_y=1}\left(q_{d_y+1}\, \hat{y}_{d_y+1} \right) \tag{A-54}$$

where t_{d_y} is given by:

$$t_{d_y} \le \sum_{d_x=1}^{D-1}\left(\hat{x}_{d_x}\, r_{d_x} \right) + \Gamma v_{d_y} \tag{A-55}$$

$$t_{d_y} \ge \sum_{d_x=1}^{D-1}\left(\hat{x}_{d_x}\, r_{d_x} \right) - \Gamma(1 - v_{d_y}) \tag{A-56}$$

$$t_{d_y} \le \hat{y}_{d_y} v_{d_y} \tag{A-57}$$

$$t_{d_y} \le \hat{x}_{d_x} r_{d_x} \tag{A-58}$$

$$t_{d_y} \ge 0 \tag{A-59}$$

Similar equations can be written for q_{d_y} :

$$q_{d_y} \le \sum_{d_x=1}^{D-1}\left(\hat{x}_{d_{x+1}}\, r_{d_x} \right) + \Gamma v_{d_y} \tag{A-60}$$

$$q_{d_y} \ge \sum_{d_x=1}^{D-1}\left(\hat{x}_{d_{x+1}}\, r_{d_x} \right) - \Gamma(1 - v_{d_y}) \tag{A-61}$$

$$q_{d_y} \le \hat{y}_{d_y} v_{d_y} \tag{A-62}$$

$$q_{d_y} \le \hat{x}_{d_x+1} r_{d_x} \tag{A-63}$$

$$q_{d_y} \ge 0 \tag{A-64}$$

Equations (A-55) through (A-58) can also be replaced by:

$$t_{d_x} \leq \hat{x}_{d_x} r_{d_x} + \Gamma(2 - v_{d_y} - r_{d_x}) \tag{A-65}$$

$$t_{d_x} \geq \hat{x}_{d_x} r_{d_x} - \Gamma(2 - v_{d_y} - r_{d_x}) \tag{A-66}$$

Similar substitutions can be made for equations (A-60) and (A-61). We omit showing the alternative equations for MCP2, which use a similar scheme than the one for MCP1.

Finally, for MCP3 we use:

$$\begin{aligned} z \geq & \hat{x}_{d_x} y + x \hat{y}_{d_y} - \hat{x}_{d_x} \hat{y}_{d_y} - \\ & -\left(\hat{x}_{d_x} \hat{y}_{d_y+1} + \hat{x}_{d_x+1} \hat{y}_{d_y} - \hat{x}_{d_x} \hat{y}_{d_y}\right)\left(2 - r_{d_x} - v_{d_y}\right) \quad \forall d_x = 1..D_x, d_y = 1..D_y \end{aligned} \tag{A-67}$$

$$\begin{aligned} z \geq & \hat{x}_{d_x+1} y + x \hat{y}_{d_y+1} - \hat{x}_{d_x+1} \hat{y}_{d_y+1} \\ & -\left(\hat{x}_{d_x+1} \hat{y}_{d_y+1} + \hat{x}_{d_x+1} \hat{y}_{d_y+1} - \hat{x}_{d_x+1} \hat{y}_{d_y+1}\right)\left(2 - r_{d_x} - v_{d_y}\right) \quad \forall d_x = 1..D_x, d_y = 1..D_y \end{aligned} \tag{A-68}$$

$$\begin{aligned} z \leq & \hat{x}_{d_x+1} y + x \hat{y}_{d_y} - \hat{x}_{d_x+1} \hat{y}_{d_y} \\ & +\left(\hat{x}_{d_x+1} \hat{y}_{d_y+1} + \hat{x}_{d_x+1} \hat{y}_{d_y}\right)\left(2 - r_{d_x} - v_{d_y}\right) \quad \forall d_x = 1..D_x, d_y = 1..D_y \end{aligned} \tag{A-69}$$

$$\begin{aligned} z \geq & \hat{x}_{d_x} y + x \hat{y}_{d_y+1} - \hat{x}_{d_x} \hat{y}_{d_y+1} + \\ & +\left(\hat{x}_{d_x+1} \hat{y}_{d_y+1} + \hat{x}_{d_x} \hat{y}_{d_y+1}\right)\left(2 - r_{d_x} - v_{d_y}\right) \quad \forall d_x = 1..D_x, d_y = 1..D_y \end{aligned} \tag{A-70}$$

$$z \leq x^U y \tag{A-71}$$

$$z \leq x y^U \tag{A-72}$$

From the few tests we performed using the direct discretization options, we observed that all the above schemes for both variables did not present real advantages. We suspect that using McCormick discretization options the results may be similar. However, a more thorough checking is needed, which we leave for future work.