

Normalized multiparametric disaggregation: an efficient relaxation for mixed-integer bilinear problems

Pedro M. Castro¹ 

Received: 20 February 2015 / Accepted: 20 July 2015 / Published online: 30 July 2015
© Springer Science+Business Media New York 2015

Abstract A key element for the global optimization of non-convex mixed-integer bilinear problems is the computation of a tight lower bound for the objective function being minimized. Multiparametric disaggregation is a technique for generating a mixed-integer linear relaxation of a bilinear problem that works by discretizing the domain of one of the variables in every bilinear term according to a numeric representation system. This can be done up to a certain accuracy level that can be different for each discretized variable so as to adjust the number of significant digits to their range of values and give all variables the same importance. We now propose a normalized formulation (NMDT) that achieves the same goal using a common setting for all variables, which is equivalent to the number of uniform partitions in a closely related, piecewise McCormick (PCM) approach. Through the solution of several benchmark problems from the literature involving four distinct problem classes, we show that the computational performance of NMDT is already better than PCM for ten partitions, with the difference rising quickly due to the logarithmic versus linear growth in the number of binary variables with the number of partitions. The results also show that a global optimization solver based on the proposed relaxation compares favorably with commercial solvers BARON and GloMIQO.

Keywords Mixed-integer nonlinear programming · Quadratic optimization · Disjunctive programming · Algorithm · Process networks

List of symbols

a_{ijq}	Scalar multiplying bilinear term $x_i x_j$ in constraint q
B_q	Matrix with coefficients for variables x in constraint q
C_q	Matrix with coefficients for variables y in constraint q
d_q	Constant term in constraint q

✉ Pedro M. Castro
pmcastro@fc.ul.pt

¹ CMAF-CIO, Faculdade de Ciências, Universidade de Lisboa, 1749-016 Lisbon, Portugal

f_0	Optimal value of objective function for problem (P)
f'_0	Optimal value of objective function for problem (P') that is equivalent to (P)
f_0^R	Optimal value of objective function for problem (PR) , lower bound for (P)
f_0^*	Upper bound for problem (P)
$f_q(\mathbf{x}, \mathbf{y})$	Function of continuous variables x and binary variables y defining constraint q
k	Digit in decimal numerical representation system, $\in \{0, \dots, 9\}$
l	Position in decimal numerical representation system, $\in \{p, \dots, -1\}$
n	Partition in piecewise McCormick relaxation, $\in \{1, \dots, N\}$
N	Number of partitions specified for piecewise McCormick relaxation
p	Parameter defining the accuracy level of discretized variables, $\in \mathbb{Z}^-$
\mathbf{x}^L	Vector of lower bounds of continuous variables x
\mathbf{x}^U	Vector of upper bounds of continuous variables x
x_i	Variable of bilinear term $x_i x_j$ to be disaggregated
\hat{x}_{ijkl}	Disaggregated variable from linearization of $x_i z_{jkl}$
\hat{x}_{ijn}	Disaggregated variable in piecewise McCormick relaxation
\hat{x}_{jn}	Disaggregated variable in piecewise McCormick relaxation
x_j	Variable of bilinear term $x_i x_j$ to be discretized
w_{ij}	Variable replacing bilinear term $x_i x_j$
z_{jkl}	Binary variable assigning to λ_j digit k to position l
z_{jn}	Binary variable assigning partition n to variable x_j
ε	Targeted relative optimality tolerance
λ_{jl}	$\lambda_j =$ Discretized variable linked to original variable x_j , $\in [0, 1]$
v_{ij}	Value of λ_j in position l of discretized representation, $\in \{0, 10^l, \dots, 9 \cdot 10^l\}$
τ	Variable replacing bilinear term $x_i \lambda_j$
τ	Parameter for comparing performance of a solver with respect to its competitors
$\Delta \lambda_j$	Slack variable ensuring continuous domain for λ_j , $\in [0, 10^p]$
Δv_{ij}	Variable replacing bilinear term $x_i \cdot \Delta \lambda_j$
$(\mathbf{x}^R, \mathbf{y}^R)$	Optimal solution for problem (PR)
$(\mathbf{x}^*, \mathbf{y}^*)$	Best-known solution for problem (P)

1 Introduction

The problem considered in this paper can be classified as a nonconvex, mixed-integer quadratically constrained problem (MIQCP) with the following general form:

$$\begin{aligned}
 & \min f_0(\mathbf{x}, \mathbf{y}) \\
 & \text{subject to} \\
 & f_q(\mathbf{x}, \mathbf{y}) \leq 0 \quad \forall q \in Q \setminus \{0\} \\
 & f_q(\mathbf{x}, \mathbf{y}) = \sum_{(i,j) \in BL} a_{ijq} x_i x_j + B_q \mathbf{x} + C_q \mathbf{y} + d_q \quad \forall q \in Q \\
 & 0 \leq \mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U \\
 & \mathbf{x} \in \mathbb{R}^m, \mathbf{y} \in \{0, 1\}^r \quad \text{(P)}
 \end{aligned}$$

where \mathbf{x} is a vector of continuous non-negative variables and \mathbf{y} are binary variables. BL is an (i, j) -index set that defines the bilinear $x_i x_j$ and quadratic terms ($i = j$) present in the problem and it is assumed that it is possible to infer finite upper bounds \mathbf{x}^U on variables x_i

and x_j . Set Q includes all functions f_q , the objective function f_0 and all the constraints, a_{ijq} and d_q are scalars whereas B_q and C_q are matrices.

Problem **(P)** is frequently encountered in engineering. An example is the optimization of process networks in process systems engineering [1]. Nodes involve equipment units that are connected by streams, which can be characterized by a flowrate and a set of properties. Whenever there are multiple input streams to a node, mixing is involved, with the properties of the output stream being estimated as a weighted sum by flowrates of the properties of the input streams. As a consequence, nonconvex bilinear terms appear.

Process network problems are considerably more general than generalized pooling problems [2,3] and include blending of crude oil [4–6] or refined products in refineries [7–9], and reducing freshwater consumption and wastewater treatment cost in industrial water networks [10–14]. Researchers started looking into the steady-state design or operation of the system, typically considering solely continuous variables. Exceptions leading to mixed-integer bilinear problems were due to constraints allowing a connecting stream only if the flowrate is above a given minimum value [3], and the selection of one of alternative technologies for a unit [10]. A recent trend made possible by the advancements in computer hardware and optimization algorithms, is the incorporation of the scheduling component through the consideration of multiple time periods [6,9]. This is also the case of the thermal unit commitment [15] and hydroelectric scheduling problems [16], which divide a 24-h horizon into hourly intervals. Note that no mixing is involved in the latter systems, with the bilinear terms appearing to compute either the production cost from a generator or the power output from a reservoir. In all mentioned cases, the binary variables appear linearly in the constraints. In contrast, the trim loss problem features bilinear terms with integer variables [17,18]. While it is not of type **(P)** the relaxation approaches discussed in this paper can also be applied [19].

Global optimization solvers [20–22] rely on linear or mixed integer linear programming relaxations of **(P)**. A tight relaxation is crucial to achieve fast convergence and this is highly dependent on the bounds of the variables involved in the bilinear terms, improving as their domain is reduced. In optimality-based bound contraction [23,24], two LPs or MILPs are solved for each such variable. Spatial branch-and-bound can also be seen as an iterative procedure acting on a single variable at a time (when branching). Simultaneous variable partitioning using piecewise McCormick envelopes [3,11,13,25–28] or multiparametric disaggregation [19,29,30] provide better approximations but may be too demanding computationally since additional binary, continuous variables and constraints, are part of the MILP relaxation.

Piecewise McCormick (PCM) involves ab initio partitioning of the domain of variables x_j in **(P)**, being uniform partitioning the most common. Since the number of binary variables added is related to the number of partitions N , this is an important tuning parameter. Experiments in [31] used $N = 2, 4$ and 8 up to a maximum of 30 partitioned variables. For a water network problem, Castro [24] proposed a formula to compute N as a function of problem size, to avoid generating intractable MILPs [26]. However, even though it increases the chances that the bilinear problem will solve to global optimality within a given time limit, the piecewise relaxation option in GloMIQO is turned off by default [20] since convergence is generally faster that way. One way to improve the quality of the piecewise McCormick relaxation for a given N is to use partition-dependent bounds also for variables x_i , which can be obtained by optimality-based bound contraction [24].

Multiparametric disaggregation (MDT) is a closely related but conceptually different type of MILP relaxation approach. It works by first discretizing variables x_j in **(P)** to a specified accuracy level p [29] and then adding slack variables to achieve continuous domains [30]. In

very special cases, the accuracy parameter of MDT can be directly related to the number of uniform partitions N of PCM with the latter allowing for a better control of the complexity of the resulting MILP. As an example, for $x_j \in [0, 1]$, N can be any integer greater than one, whereas the full set of values for $p \in \{\dots, -3, -2, -1\}$ corresponds to the subset $N \in \{10, 100, 1000, \dots\}$. Accuracy in MDT can thus only change by one order of magnitude since partitioning is based on a numeric representation system. Yet, for MDT, the number of added binary variables grows logarithmically with N rather than linearly (for a common implementation of piecewise McCormick [30,32], other piecewise linear schemes are not as harsh [33]). As a consequence, MDT may reach considerably smaller optimality gaps, due to a computational performance that is often orders of magnitude faster [32].

In the context of a spatial branch-and-bound algorithm with frequent domain reduction, PCM has another advantage over MDT. Besides also benefiting from tighter variable bounds, it can generate consistently shorter partitions while keeping the value of N constant. In contrast, MDT will require fewer discrete points (i.e. fewer binary variables) to include the reduced domain, but the location of those still required will be exactly the same. In other words, we have with PCM a double impact in relaxation quality with the same problem size whereas in MDT we have a smaller impact with a reduced size. Since the ultimate goal is to prove global optimality, the former is preferable.

In this paper, we bring multiparametric disaggregation closer to uniform piecewise McCormick. The main novelty is to consider a dimensionless domain for the discretized variables in MDT. More specifically, rather than discretizing all possible values that a variable can assume, we discretize the range $[0, 1]$ between the lower (LB) and upper bound (UB). One immediate consequence is that the accuracy level parameter p can now be directly related to the number of partitions N , regardless of the LB and UB values, allowing for a better comparison between the two alternative relaxation methods for (P). Note that its most important feature of scaling logarithmically with the number of partitions is kept.

2 Normalized multiparametric disaggregation

Given a nonconvex bilinear term $w_{ij} = x_i x_j$, multiparametric disaggregation works by discretizing x_j over a set of powers $l \in \{p, \dots, P\}$, where $P = \lfloor \log_{10} x_j^U \rfloor$ and p is chosen by the user so as to reach a certain accuracy level [23,29,30]. The formula for P assumes discretization with base 10, preferred in this paper over bases 2 and 16 because it makes the approach easier to describe and illustrate (the interested reader can find in references [9,19,23,32,34] the full set of constraints for a generic base and extensive computational results for the binary representation system).

We now propose a normalized version of multiparametric disaggregation that discretizes $\lambda_j \in [0, 1]$, an auxiliary variable that is used to compute x_j as a linear combination of its lower x_j^L and upper x_j^U bounds:

$$x_j = x_j^L + \lambda_j (x_j^U - x_j^L) \quad \forall j \quad (1)$$

The exact representation of λ_j can be achieved by considering an infinite number of positions $l \in \mathbb{Z}^-$ in the decimal system,

$$\lambda_j = \sum_{l \in \mathbb{Z}^-} \lambda_{jl} \quad \forall j \quad (2)$$

and by picking the appropriate digit $k \in \{0, 1, \dots, 9\}$ for each power l . This can be stated as a disjunction [35,36], where binary variables z_{jkl} take the value of one if digit k is selected for position l for discretized variable λ_j :

$$\bigvee_{k=0}^9 \left[\lambda_{jl} = 10^l \cdot k \right] \quad \forall j, l \in \mathbb{Z}^- \tag{3}$$

The convex hull reformulation [37] of the disjunction in (3) can be simplified so as to generate a sharp formulation without disaggregated variables [38].

$$\lambda_j = \sum_{l \in \mathbb{Z}^-} \sum_{k=0}^9 10^l \cdot k \cdot z_{jkl} \quad \forall j \tag{4}$$

$$\sum_{k=0}^9 z_{jkl} = 1 \quad \forall j, l \in \mathbb{Z}^- \tag{5}$$

Multiplying variable x_i by (1) and substituting $x_i x_j$ and $x_i \lambda_j$ with bilinear variables w_{ij} and v_{ij} leads to,

$$w_{ij} = x_i x_j^L + v_{ij} (x_j^U - x_j^L) \quad \forall (i, j) \tag{6}$$

Substituting (4) into the definition of v_{ij} leads to the appearance of bilinear terms involving the product of a continuous and a binary variable.

$$v_{ij} = \sum_{l \in \mathbb{Z}^-} \sum_{k=0}^9 10^l \cdot k \cdot x_i z_{jkl} \quad \forall (i, j) \tag{7}$$

We can now perform an exact linearization [39] by introducing new continuous variables $\hat{x}_{ijkl} = x_i z_{jkl}$ so that:

$$v_{ij} = \sum_{l \in \mathbb{Z}^-} \sum_{k=0}^9 10^l \cdot k \cdot \hat{x}_{ijkl} \quad \forall (i, j) \tag{8}$$

$$z_{jkl} x_i^L \leq \hat{x}_{ijkl} \leq z_{jkl} x_i^U \quad \forall (i, j), k \in \{0, \dots, 9\}, l \in \mathbb{Z}^- \tag{9}$$

Finally, multiplying (5) by x_i and replacing the bilinear terms by the new continuous variables results in,

$$x_i = \sum_{k=0}^9 \hat{x}_{ijkl} \quad \forall (i, j), l \in \mathbb{Z}^- \tag{10}$$

The full set of mixed integer linear constraints for the exact representation of bilinear terms $w_{ij} = x_i x_j$ is thus given by Eqs. (1), (4–6) and (8–10), leading to optimization problem (P’).

$$\min f'_0(\mathbf{x}, \mathbf{y})$$

subject to

$$f'_q(\mathbf{x}, \mathbf{y}) \leq 0 \quad \forall q \in Q \setminus \{0\}$$

$$f'_q(\mathbf{x}, \mathbf{y}) = \sum_{(i,j) \in BL} a_{ijq} w_{ij} + B_q \mathbf{x} + C_q \mathbf{y} + d_q \quad \forall q \in Q$$

$$\left. \begin{aligned} x_j &= x_j^L + \lambda_j (x_j^U - x_j^L) \\ \lambda_j &= \sum_{l \in \mathbb{Z}^-} \sum_{k=0}^9 10^l \cdot k \cdot z_{jkl} \end{aligned} \right\} \quad \forall j \in \{j \mid (i, j) \in BL\}$$

$$\left. \begin{aligned}
 w_{ij} &= x_i x_j^L + v_{ij} (x_j^U - x_j^L) \\
 v_{ij} &= \sum_{l \in \mathbb{Z}^-} \sum_{k=0}^9 10^l \cdot k \cdot \hat{x}_{ijkl}
 \end{aligned} \right\} \quad \forall (i, j) \in BL$$

$$x_i = \sum_{k=0}^9 \hat{x}_{ijkl} \quad \forall (i, j) \in BL, l \in \mathbb{Z}^-$$

$$\sum_{k=0}^9 z_{jkl} = 1 \quad \forall j \in \{j | (i, j) \in BL\}, l \in \mathbb{Z}^-$$

$$z_{jkl} x_i^L \leq \hat{x}_{ijkl} \leq z_{jkl} x_i^U \quad \forall (i, j), k \in \{0, \dots, 9\}, l \in \mathbb{Z}^-$$

$$0 \leq \mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U$$

$$\mathbf{x} \in \mathbb{R}^m, w_{ij}, \lambda_j, v_{ij}, \hat{x}_{ijkl} \in \mathbb{R}$$

$$\mathbf{y} \in \{0, 1\}^r, z_{jkl} \in \{0, 1\} \quad (\mathbf{P}')$$

The following property can be readily established from the derivation of (P').

Property 1 Problem (P') is equivalent to problem (P), i.e. $f'_0 = \min_{\mathbf{x}, \mathbf{y}} f'_0(\mathbf{x}, \mathbf{y}) = \min_{\mathbf{x}, \mathbf{y}} f_0(\mathbf{x}, \mathbf{y}) = f_0$.

2.1 Lower bounding formulation

Since it is impossible to compute the infinite sums over all negative integers, we represent λ_j to a finite accuracy level by replacing $l \in \mathbb{Z}^-$ with $l \in \{p, p + 1, \dots, -1\}$, where p is a negative integer chosen by the user. In order to close the gap between discretization points so as to allow for all possible values for λ_j , we introduce slack variables $\Delta\lambda_j$ that are bounded between 0 and 10^p , see Fig. 1. The continuous representation of λ_j is then given by:

$$\lambda_j = \sum_{l=p}^{-1} \sum_{k=0}^9 10^l \cdot k \cdot z_{jkl} + \Delta\lambda_j \quad \forall j \tag{11}$$

$$0 \leq \Delta\lambda_j \leq 10^p \quad \forall j \tag{12}$$

As an example, to obtain $\lambda_j = 0.4385$ for $p = -2$, we need $z_{j,4,-1} = 1, z_{j,3,-2} = 1$ and $\Delta\lambda_j = 0.0085 \leq 0.01 = 10^{-2}$. To generate $\lambda_j = 1$ for $p = -1$, the non-zero variables are $z_{j,9,-1} = 1$ and $\Delta\lambda_j = 0.1$.

For the continuous representation of the bilinear term $v_{ij} = x_i \lambda_j$, and following the same reasoning as before, we get:

$$v_{ij} = \sum_{l=p}^{-1} \sum_{k=0}^9 10^l \cdot k \cdot \hat{x}_{ijkl} + x_i \cdot \Delta\lambda_j \quad \forall (i, j) \tag{13}$$

Notice the appearance of new bilinear terms $x_i \cdot \Delta\lambda_j$ that are going to be relaxed using the McCormick envelopes [40], which in this case coincide with the reformulation linearization

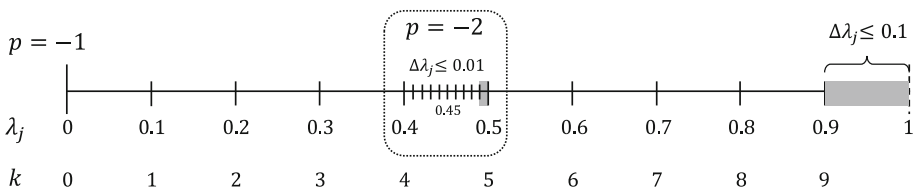


Fig. 1 The continuous representation of variable λ_j is achieved by discretizing the domain up to a certain level p and adding a bounded variable $\Delta\lambda_j$

technique bound factor products $x_i - x_i^L \geq 0, x_i^U - x_i \geq 0, \Delta\lambda_j \geq 0$ and $10^p - \Delta\lambda_j \geq 0$ [41,42]. Variables Δv_{ij} replace $x_i \cdot \Delta\lambda_j$ in Eq. (13).

$$x_i^L \cdot \Delta\lambda_j \leq \Delta v_{ij} \leq x_i^U \cdot \Delta\lambda_j \quad \forall (i, j) \tag{14}$$

$$(x_i - x_i^U) \cdot 10^p + x_i^U \cdot \Delta\lambda_j \leq \Delta v_{ij} \leq (x_i - x_i^L) \cdot 10^p + x_i^L \cdot \Delta\lambda_j \quad \forall (i, j) \tag{15}$$

Replacing Eqs. (4) and (8) in (P*) by (11–15), we obtain the new optimization problem (PR) that is a relaxation of (P). In other words, (PR) is feasible for values of w_{ij}, x_i and x_j that do not satisfy $w_{ij} = x_i x_j$.

min $f_0^R(\mathbf{x}, \mathbf{y})$

subject to

$$f_q^R(\mathbf{x}, \mathbf{y}) \leq 0 \quad \forall q \in Q \setminus \{0\}$$

$$f_q^R(\mathbf{x}, \mathbf{y}) = \sum_{(i,j) \in BL} a_{ijq} w_{ij} + B_q \mathbf{x} + C_q \mathbf{y} + d_q \quad \forall q \in Q$$

$$\left. \begin{aligned} x_j &= x_j^L + \lambda_j (x_j^U - x_j^L) \\ \lambda_j &= \sum_{l=p}^{-1} \sum_{k=0}^9 10^l \cdot k \cdot z_{jkl} + \Delta\lambda_j \\ 0 &\leq \Delta\lambda_j \leq 10^p \end{aligned} \right\} \quad \forall j \in \{j \mid (i, j) \in BL\}$$

$$\left. \begin{aligned} w_{ij} &= x_i x_j^L + v_{ij} (x_j^U - x_j^L) \\ v_{ij} &= \sum_{l=p}^{-1} \sum_{k=0}^9 10^l \cdot k \cdot \hat{x}_{ijkl} + \Delta v_{ij} \\ x_i^L \cdot \Delta\lambda_j &\leq \Delta v_{ij} \leq x_i^U \cdot \Delta\lambda_j \\ \Delta v_{ij} &\leq (x_i - x_i^L) \cdot 10^p + x_i^L \cdot \Delta\lambda_j \\ \Delta v_{ij} &\geq (x_i - x_i^U) \cdot 10^p + x_i^U \cdot \Delta\lambda_j \end{aligned} \right\} \quad \forall (i, j) \in BL$$

$$x_i = \sum_{k=0}^9 \hat{x}_{ijk} \quad \forall (i, j) \in BL, l \in \{p, \dots, -1\}$$

$$\sum_{k=0}^9 z_{jkl} = 1 \quad \forall j \in \{j \mid (i, j) \in BL\}, l \in \{p, \dots, -1\}$$

$$z_{jkl} x_i^L \leq \hat{x}_{ijkl} \leq z_{jkl} x_i^U \quad \forall (i, j), k \in \{0, \dots, 9\}, l \in \{p, \dots, -1\}$$

$$0 \leq \mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U$$

$$\mathbf{x} \in \mathbb{R}^m, w_{ij}, \lambda_j, v_{ij}, \hat{x}_{ijkl}, \Delta\lambda_j, \Delta v_{ij} \in \mathbb{R}$$

$$\mathbf{y} \in \{0, 1\}^r, z_{jkl} \in \{0, 1\} \tag{PR}$$

The following property can be stated from the above derivation and discussion:

Property 2 The solution of problem (PR) yields a lower bound for problem (P), i.e. $f_0^R = \min_{\mathbf{x}, \mathbf{y}} f_0^R(\mathbf{x}, \mathbf{y}) \leq f_0$.

2.2 Algorithm for global optimization

All algorithms for global optimization work by approaching a lower bound from the solution of a relaxation problem to an upper bound typically obtained from solving a constrained version of the original problem. Spatial branch and bound is an iterative procedure for raising the lower bound that acts on a single variable at a time. In contrast, the proposed MILP relaxation

(PR) of the original MINLP problem (P) involves simultaneous variable partitioning and so can be made as tight as desired simply by acting on accuracy level parameter p .

Property 3 As p approaches $-\infty$, f_0^R approaches f_0 .

Proof As p approaches $-\infty$ in (PR) we get from Eq. (12):

$$\lim_{p \rightarrow -\infty} \Delta\lambda_j = \lim_{p \rightarrow -\infty} 10^p = 0$$

Then, from Eq. (14):

$$\begin{aligned} \lim_{p \rightarrow -\infty} \Delta v_{ij} &= \lim_{p \rightarrow -\infty} (x_i^L \cdot \Delta\lambda_j) = \lim_{p \rightarrow -\infty} (x_i^U \cdot \Delta\lambda_j) \\ &= x_i^U \cdot \lim_{p \rightarrow -\infty} \Delta\lambda_j = x_i^U \cdot 0 = 0 \end{aligned}$$

And from Eq. (15):

$$\begin{aligned} \lim_{p \rightarrow -\infty} \Delta v_{ij} &= \lim_{p \rightarrow -\infty} [(x_i - x_i^U) \cdot 10^p + x_i^U \cdot \Delta\lambda_j] \\ &= \lim_{p \rightarrow -\infty} [(x_i - x_i^L) \cdot 10^p + x_i^L \cdot \Delta\lambda_j] \\ &= (x_i - x_i^L) \cdot \lim_{p \rightarrow -\infty} 10^p + \lim_{p \rightarrow -\infty} (x_i^L \cdot \Delta\lambda_j) \\ &= (x_i - x_i^L) \cdot 0 + 0 = 0 \end{aligned}$$

Variables $\Delta\lambda_j$ and Δv_{ij} are thus eliminated. Since the domain of l in the summations and constraints of (PR) becomes the set of negative integers \mathbb{Z}^- , (PR) becomes (P') and hence f_0^R approaches f_0' . Then, from Property 1, f_0^R approaches f_0 . □

The optimal solution from (PR) provides a good initialization point for the solution of (P). An efficient method to generate a feasible solution for (P) that will be an upper bound, is to fix all binary variables and solve the resulting NLP with a fast local solver [30,32]. Note however, that this may render the problem infeasible, which is more likely to occur for higher p values.

The global optimization algorithm resulting from the aforementioned lower and upper bounding methods is as follows. Starting with the lowest accuracy level, we solve (PR) and (P) in sequence. If the difference between the values of the objective function is smaller than the given relative optimality tolerance ε , then the algorithm terminates; otherwise, accuracy is increased to the next level and the problems are solved once more.

Algorithm

Step 0. Choose $p = -1$ and let $f_0^* = +\infty$.

Step 1. Solve (PR) to obtain f_0^R and point (x^R, y^R) .

Step 2. Add constraint $y = y^R$ to (P) reducing it to an NLP. Using x^R as a starting point, solve the NLP with a local solver. If the NLP is feasible and a lower value is obtained for the objective function, update the upper bound f_0^* and best solution (x^*, y^*) .

Step 3. If $(f_0^* - f_0^R) / f_0^R \leq \varepsilon$, STOP, the solution (x^*, y^*) is globally optimal. Otherwise, set $p = p - 1$ and return to step 1.

3 Motivation for normalized multiparametric disaggregation

The difference between the standard and normalized multiparametric disaggregation comes down to the choice of either discretizing the values of the variables or the range between their lower and upper bounds. Discretized variables will typically have dissimilar domains, perhaps even of a different order of magnitude, and it is difficult to predict which will lead to the largest error between the exact representation of the bilinear term and its mixed-integer linear relaxation. While it is straightforward to alter the formulations to consider a variable dependent accuracy parameter p , given the large number of discretized variables that may be involved it is highly desirable to keep the number of tuning parameters to a minimum and consider a global value instead.

If this is done for the standard MDT, we are effectively giving a different importance to the discretized variables, as can be seen in the simple example in Fig. 2. Assuming accuracy in the units level ($p = 1$), MDT generates 11 discrete points for variable $x_1 \in [0, 10]$, whereas variable $x_2 \in [3, 5]$ gets only three. In contrast, our new NMDT approach generates the same relaxation for x_1 using $p = -1$, while x_2 gets the same number of discrete points as x_1 . Compared to MDT, variable x_2 is now partitioned in uniform intervals that are five times shorter, potentially leading to a tighter relaxation.

The normalized multiparametric disaggregation can also take further advantage of information resulting from bound contracting procedures that are frequently called in spatial branch and bound algorithms, as can be seen from the solution of problem (P1). This has been adapted from Problem 106 in Hock and Schittowski [43] so as to facilitate comparison. More specifically, the lower bounds for variables x_4 to x_8 have been lowered from 10 to 0, but this has no effect in the value of the objective function: 7049.248. Of the different choices available concerning the discretized variables, we select to discretize x_4, \dots, x_8 .

$$\begin{aligned}
 &\min x_1 + x_2 + x_3 \\
 &\text{s.t. } 0.0025(x_4 + x_6) - 1 \leq 0 \\
 &\quad 0.0025(-x_4 + x_5 + x_7) - 1 \leq 0 \\
 &\quad 0.01(-x_5 + x_8) - 1 \leq 0 \\
 &\quad 100x_1 - x_1x_6 + 833.33252x_4 - 83,333.333 \leq 0 \\
 &\quad x_2x_4 - x_2x_7 - 1250x_4 + 1250x_5 \leq 0 \\
 &\quad x_3x_5 - x_3x_8 - 2500x_5 + 1,250,000 \leq 0 \\
 &\quad 100 \leq x_1 \leq 10,000, 1000 \leq x_2, x_3 \leq 10,000, 0 \leq x_4, \dots, x_8 \leq 1000
 \end{aligned} \tag{P1}$$

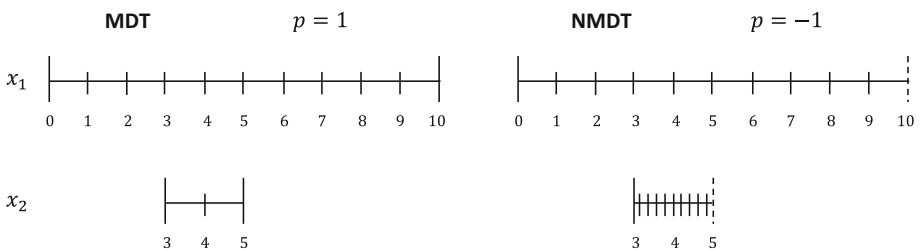


Fig. 2 Unlike the standard approach (MDT), the normalized multiparametric disaggregation (NMDT) generates the same number of discrete points for all discretized variables using a global accuracy parameter p

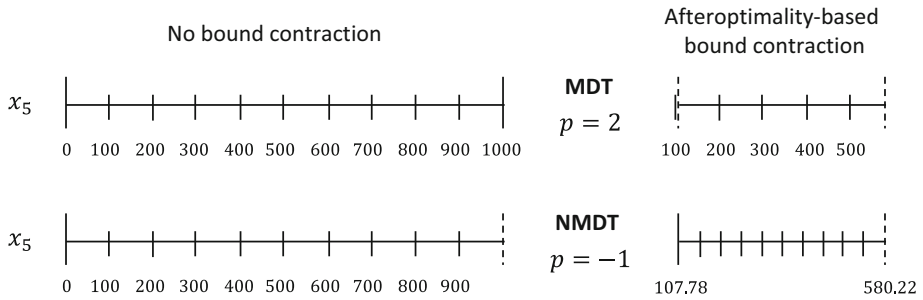


Fig. 3 Discrete points generated for variable x_5 of (P1) by standard (MDT) and normalized multiparametric disaggregation (NMDT) for the cases of original and tightened variable bounds

Table 1 Comparison between standard (MDT) and normalized multiparametric disaggregation (NMDT) for problem (P1)

Relaxation preceded by bound contraction?		No	Yes	No	Yes	No	Yes
Approach	p	Gap (%)		Binary variables		CPUs	
MDT	2	56.1	55.1	60	26	0.38	0.25
NMDT	-1	56.1	28.1	50	50	0.38	0.37
MDT	1	6.95	3.39	110	76	1.13	0.73
NMDT	-2	6.95	1.66	100	100	1.15	1.82
MDT	0	0.711	0.268	160	126	5.36	5.25
NMDT	-3	0.711	0.156	150	150	6.41	10.7
MDT	-1	0.0713	0.0261	210	176	81.7	41.9
NMDT	-4	0.0713	0.0160	200	200	49.4	133
MDT	-2	0.0071	0.0026	260	226	1172	756
NMDT	-5	0.0071	0.0016	250	250	626	2736

Assume for illustrative purposes that discrete points are placed at the hundreds, which corresponds to a coarse accuracy level, as can be seen in Fig. 3 for variable x_5 . It corresponds to $p = -1$ for discretized variable λ_5 in NMDT and to $p = 2$ for discretized variable x_5 in MDT [30], leading respectively to the definition of 10 and 12 binary variables for this variable. The two extra binaries for MDT result from the way it was coded, and is due to discrete point 1000 that is part of the domain of x_5 . In contrast, in NMDT, $x_5 = 1000$ is achieved by making $z_{5,9,-1} = 1$ and $\Delta\lambda_5 = 0.1$, recall Eq. (11). Despite this minor difference, the quality of the relaxation is the same for both approaches.

If we now apply optimality based bound contraction [23] using the McCormick envelopes [40] to generate the relaxation, the domain can almost be cut by half to $x_5 \in [107.78, 580.22]$. As a consequence, MDT reduces the number of discrete points to five, with the first ($x_5 = 100$) being actually placed outside the variable domain so as to obtain values up to 200 by adding the contribution of the slack variable. Due to the tighter bounds, the relative optimality gap, $gap = \frac{f_0 - f_0^R}{f_0^R}$, decreases by a factor between 1.02 and 2.74 from $p = 2$ to $p = -2$ (Table 1). In contrast, NMDT keeps the same ten discrete points, but since these are now closer together, the optimality gap can be reduced further by a factor between 2.00 ($p = -1$) and 4.48 ($p = -5$).

With the increase in problem size, it becomes apparent from the results in Table 1 that the relaxation problems generated from the new approach can be solved faster. However, this is no longer the case if their solution is preceded by bound contraction. In fact, there is a reduction in computational time for MDT, which was expected due to the fewer binary variables, but for NMDT the effort typically increases, reaching a factor of 4.37 for $p = -5$. Thus, there is a clear tradeoff between relaxation quality and computational effort.

4 Piecewise McCormick relaxation

Piecewise McCormick envelopes [2, 11, 25] also provide a MILP relaxation for problem (P). Of at least 15 alternatives that have been proposed [26, 27], we consider the one below that relies on uniform partitioning, noting that others may perform better. The basic principle is to divide the domain of variable x_j in bilinear term $w_{ij} = x_i x_j$ into N partitions and apply the McCormick envelopes [40] on each, taking advantage that the partition bounds x_{jn}^L and x_{jn}^U are tighter than the global bounds x_j^L and x_j^U , see Fig. 4. Binary variables z_{jn} select the best partition.

The complete lower bounding formulation (PR-PCM) is given below.

$$\begin{aligned}
 & \min f_0^R(\mathbf{x}, \mathbf{y}) \\
 & \text{subject to} \\
 & f_q^R(\mathbf{x}, \mathbf{y}) \leq 0 \qquad \qquad \qquad \forall q \in Q \setminus \{0\} \\
 & f_q^R(\mathbf{x}, \mathbf{y}) = \sum_{(i,j) \in BL} a_{ijq} w_{ij} + B_q \mathbf{x} + C_q \mathbf{y} + d_q \qquad \forall q \in Q \\
 & \left. \begin{aligned}
 w_{ij} & \geq \sum_{n=1}^N \left(\hat{x}_{ijn} x_{jn}^L + \hat{x}_{jn} x_i^L - z_{jn} x_i^L x_{jn}^L \right) \\
 w_{ij} & \geq \sum_{n=1}^N \left(\hat{x}_{ijn} x_{jn}^U + \hat{x}_{jn} x_i^U - z_{jn} x_i^U x_{jn}^U \right) \\
 w_{ij} & \leq \sum_{n=1}^N \left(\hat{x}_{ijn} x_{jn}^L + \hat{x}_{jn} x_i^U - z_{jn} x_i^U x_{jn}^L \right) \\
 w_{ij} & \leq \sum_{n=1}^N \left(\hat{x}_{ijn} x_{jn}^U + \hat{x}_{jn} x_i^L - z_{jn} x_i^L x_{jn}^U \right) \\
 x_i & = \sum_{n=1}^N \hat{x}_{ijn}
 \end{aligned} \right\} \qquad \forall (i, j) \in BL \\
 & \left. \begin{aligned}
 x_j & = \sum_{n=1}^N \hat{x}_{jn} \\
 \sum_{n=1}^N z_{jn} & = 1
 \end{aligned} \right\} \qquad \forall j \in \{j \mid (i, j) \in BL\} \\
 & z_{jn} x_i^L \leq \hat{x}_{ijn} \leq z_{jn} x_i^U \qquad \qquad \qquad \forall (i, j), n \in \{1, \dots, N\} \\
 & \left. \begin{aligned}
 z_{jn} x_{jn}^L & \leq \hat{x}_{jn} \leq z_{jn} x_{jn}^U \\
 x_{jn}^L & = x_j^L + \left(x_j^U - x_j^L \right) \cdot (n - 1) / N \\
 x_{jn}^U & = x_j^U + \left(x_j^U - x_j^L \right) \cdot n / N
 \end{aligned} \right\} \qquad \forall j \in \{j \mid (i, j) \in BL\}, n \in \{1, \dots, N\} \\
 & 0 \leq \mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U \\
 & \mathbf{x} \in \mathbb{R}^m, w_{ij}, \hat{x}_{jn}, \hat{x}_{ijn} \in \mathbb{R} \\
 & \mathbf{y} \in \{0, 1\}^r, z_{jn} \in \{0, 1\} \qquad \qquad \qquad \text{(PR - PCM)}
 \end{aligned}$$

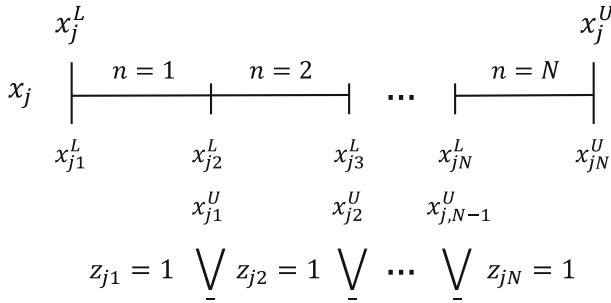


Fig. 4 In the piecewise McCormick approach, variable x_j is divided into N partitions, with binary variables z_{jn} selecting the optimal one

5 Normalized multiparametric disaggregation versus piecewise McCormick

The new normalized multiparametric disaggregation approach can also be viewed as partitioning the domain of variable x_j . However, every single decrease in p corresponds to a ten-fold increase in N .

Remark 1 The partitioning (accuracy) level parameters N , of the piecewise McCormick, and p , of normalized multiparametric disaggregation technique, are related as follows: $N = 10^{-p}$.

Piecewise McCormick is thus better at controlling the tightness of the MILP relaxation of the original MINLP problem, which may be an important point when dealing with large bilinear problems. The advantage of multiparametric disaggregation is that it scales much more favorably with the accuracy level [30,32].

Remark 2 The number of binary variables required per partitioned variable x_j is equal to N for piecewise McCormick formulation (**PR-PCM**), and to $10 \log N$ for normalized multiparametric disaggregation.

From Remarks 1 and 2, one can claim that piecewise McCormick is the only option for $N \in \{2, \dots, 9\}$ and predict that normalized multiparametric disaggregation will have a stronger computational performance for $N \in \{100, 1000, \dots\}$. But which approach is better when the same number of binary variables is involved ($N = 10$)? This question will be answered in Sect. 6.1.

5.1 The special case of quadratic terms

Previous research has shown [30] that if (**P**) is restricted to strictly bilinear terms, the quality of the relaxation from multiparametric disaggregation is the same as from piecewise McCormick (for the same accuracy level). This is no longer the case if quadratic terms are involved, with the latter approach leading to a tighter relaxation.

For illustration purposes, consider the very simple nonlinear problem (**P2**) taken from [44], for which the global optimal solution is equal to 58.383672 at (2.555772, 3.130169).

$$\begin{aligned} \min & 6x_1^2 + 4x_2^2 - 2.5x_1x_2 \\ \text{s.t.} & x_1x_2 - 8 \geq 0 \end{aligned}$$

Table 2 Value of objective function f_0^R for motivating example **(P2)** with quadratic terms

Relaxation approach	Partitions (N)		
	10	100	1000
Piecewise McCormick (PR-PCM)	46.96421	57.25218	58.27026
Normalized multiparametric disaggregation (PR)	43.25053	56.83288	58.19587

$$1 \leq x_1, x_2 \leq 10 \tag{P2}$$

As can be seen in Table 2, the value of the lower bound from **(PR-PCM)** is always higher than from **(PR)**, with the absolute difference decreasing with an increase in the number of partitions.

6 Numerical experiments

The performance of the new normalized multiparametric disaggregation approach is evaluated through the solution of 43 benchmark problems from the literature. These involve four distinct sets of process network problems of type **(P)** featuring solely strictly bilinear terms. More specifically, we have: (i) 18 water-using network design problems [34]: WUN_2-18 and 20; (ii) 15 distributed wastewater treatment networks [34]: WTN_2-16; (iii) 7 multi-period blending problems [9]: MPBP_029, 146, 480, 531, 718, 721, 852; (iv) 3 hydroelectric scheduling problems considering a subset of reservoirs [23]: HYD_2, 4 and 7.

In the given problems, bilinear terms involve the multiplication of two different sets of variables: flowrates by concentrations in (i)–(iii) and flowrates by volumes in (iv). Of the two alternatives available to generate the relaxation MILP, we choose to discretize concentrations in (i)–(iii) and flowrates in (iv). Note that problem types (i)–(ii) are NLPs, whereas (iii)–(iv) are MINLPs. The latter are freely available in www.minlp.org and also include the model for the standard multiparametric disaggregation approach. Statistics related to problem size and number of nonlinear terms for problems (i)–(iii) can be found in [45] since all these problems are part of the GloMIQO 2.2 [20] test suite.

All mathematical formulations and algorithms were implemented in GAMS 24.3 and solved on an Intel i7-4790 (3.6 GHz) processor with 8 GB of RAM, solid-state drive and running Windows 7, 64-bit operating system. The MILP relaxation problems **(PR)** and **(PR-PCM)** were solved by CPLEX 12.6 running in parallel deterministic mode using up to 8 threads. The termination criteria were a relative optimality tolerance equal to 10^{-6} or a maximum computational effort equal to 3600 CPUs for problems (i)–(iii) and 18,000 CPUs for the most difficult problems (iv). The NLPs arising from either the complete problem **(P)** or its constrained version without the binary variables (recall step 2 of the algorithm presented in Sect. 2.2), were tackled by local optimization solver CONOPT 3.16C [46]. Problem **(P)** was also tackled by global optimization solvers BARON 14.0.3 [21, 47] and GloMIQO 2.3 [20], a MIQCP standalone solver that is also part of the general MINLP solver ANTIGONE [22].

Optimality-based bound contraction, illustrated in Fig. 3, was used exclusively in problem **(P1)**.

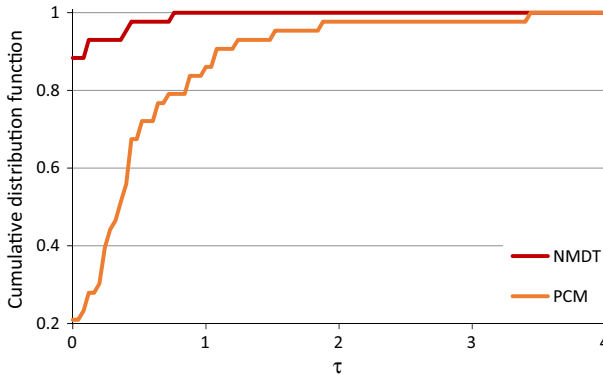


Fig. 5 Performance profile for computational time when considering 10 partitions for all discretized variables

6.1 Which approach is best for 10 partitions?

We start by solving the MILP relaxation problems from normalized multiparametric disaggregation (NMDT) and piecewise McCormick (PCM), using respectively $p = -1$ and $N = 10$ in all 43 benchmark problems. We know from the discussion in Sect. 5 that the MILPs from NMDT and PCM feature the same number of binary variables (the number of total variables and constraints is very similar) and give rise to the exact same relaxation f_0^R , and so the computational time is the obvious choice for performance measure. To get the overall assessment of the performance of a solver compared to the other solvers in a test group, one can generate the performance profiles of Dolan and Moré [48], a widely used tool for benchmarking and comparing optimization software. We plot the cumulative distribution as a function of τ , a parameter that tells us that the performance ratio of a solver with respect to the best solver is below 2^τ . As an example, the results for $\tau = 0$ give the probability that a solver is the fastest, while for $\tau = 1$ we have the probability that a solver is at most 2 times slower than the fastest solver.

From the results in Fig. 5, one can see that the new approach is clearly superior, being the fastest 88.3 % of the times vs. 20.9 % for PCM (note that ties count for both solvers and so the sum can be greater than one). Nevertheless, it should be highlighted that the computational times are typically of the same order of magnitude, the most notable exception being WTN_9, which took 1360 CPUs to solve to optimality by NMDT and 14,436 CPUs by PCM. It is the reason why the two curves meet around $\tau = 3.4$ ($2^{3.4} \approx \frac{14,436}{1360}$).

The above comparison is important in the context of convex hull reformulations of generalized disjunctive programming (GDP) models [24, 35–37, 49]. While for multiparametric disaggregation there is no need to define disaggregated variables associated to discretized variable x_j [as discussed in Sect. 2, Eqs. (3–4)], these can be found in (PR-PCM) in the form of continuous variables \hat{x}_{jn} . We do require disaggregated variables \hat{x}_{ijkl} in (PR), which for $N = 10$ have a direct correspondence with \hat{x}_{ijn} in (PR-PCM). Overall, when alternatives exist to model a particular problem as a GDP, one should look for the one leading to the most compact convex hull reformulation, since this will typically lead to the best performance, see also [49].

6.2 Comparison to commercial solvers

The algorithm discussed in Sect. 2.2 for normalized multiparametric disaggregation is capable of finding feasible solutions to problem (P) and proving global optimality if sufficient com-

putational resources are provided. It has also been implemented for piecewise McCormick, after replacing $p = -1$ with $N = 10$ in step 0, **(PR)** with **(PR-PCM)** in step 1, and $p = p - 1$ with $N = 10N$ in step 3 [32]. Note that it has been verified experimentally that for a particular iteration both NMDT and PCM lead to the same value for the objective function f_0^R , if enough time is given for the MILP solver (CPLEX) to converge. The total computational time listed is the sum over all iterations and does not exceed the maximum limits defined in Sect. 6. Since the focus is on relaxation quality, the value of f_0^* used to compute the optimality gap was the best known solution and not necessarily the best value found in step 2. The same is also true for the outputs of commercial global optimization solvers BARON and GloMIQO that are used for comparative purposes.

Table 3 provides an overview of computational performance by listing key performance indicators. While one can see that a particular algorithm is the best performer in at least two problems, the new method and GloMIQO have the most wins. Note that the first criteria used for this metric was the optimality gap, ultimately leading to solutions that are proven global optimal, followed by computational time. GloMIQO actually proved global optimality the most, while BARON was the most successful in finding the actual optima. All algorithms returned worse solutions in HYD_4 and HYD_7 than the outer approximation solver DICOPT [50] (the values in column 2 of Table 4 are taken from [23]). The other two failures for NMDT and GloMIQO came from the multiperiod blending problems. Interestingly, BARON could solve all multiperiod blending problems in less than half an hour (see detailed results in Table 4). On the other hand, and considering the comparison between the commercial solvers, GloMIQO did a better job solving the NLPs resulting from the two types of water network problems.

Another aspect worth highlighting from the results in Table 3, is related to the ability of the NMDT algorithm to find optimal solutions. Note that unlike the commercial algorithms that solve constrained versions of **(P)** in many nodes of the spatial branch and bound tree, the new algorithm solves **(P)** just a few times but still manages to find the optima. This is a direct consequence of having a tight relaxation that provides excellent starting points for the solution of **(P)** with a local solver (recall step 2 in Sect. 2.2) and is consistent with our previous work. In that sense, and due to the worse scaling of problem size, PCM cannot go to accuracy levels as deep as NMDT (see p values in Table 4), returning lower f_0^R values and inferior initialization points that sometimes lead to failure.

While the key performance indicators are useful, they do not tell the whole story. The detailed results in Table 4 are needed for a more thorough analysis, which is now performed

Table 3 Key performance indicators for the different global optimization algorithms

Algorithm	Optimal solutions		Suboptimal solutions	No solutions	Best performer
	Proven	Found			
Normalized multiparametric disaggregation (NMDT)	21	39	4	0	21
GloMIQO	23	39	4	0	18
BARON	17	41	2	0	2
Piecewise McCormick (PCM)	12	34	8	1	2
				Total	43

Table 4 Detailed results for the solution of the benchmark NLP and MINLP problems

Problem	Optimum	Accuracy p		TCPU s			Optimality gap (%)				
		PCM	NMDT	PCM	NMDT	BARON	GloMIQO	PCM	NMDT	BARON	GloMIQO
WUN_2	74.46994	-2	-4	MRL	MRL	MRL	MRL	0.2156	0.0021	0.4518	0.2873
WUN_3	143.4126	-2	-4	MRL	MRL	MRL	MRL	0.0098	0.001	0.2548	0.4217
WUN_4	123.9286	-1	-1	0.74	0.6	MRL	MRL	GO	GO	0.1845	0.0956
WUN_5	197.6901	-1	-1	0.5	0.38	39.5	0.23	GO	GO	GO	GO
WUN_6	142.0816	-2	-3	MRL	MRL	MRL	MRL	0.0644	0.0045	0.2229	0.4932
WUN_7	280.7712	-2	-4	MRL	MRL	MRL	16	0.5457	0.0056	0.0233	GO
WUN_8	164.4898	-3	-4	MRL	MRL	145	7.64	0.1277	0.0109	GO	GO
WUN_9	312.9215	-1	-3	MRL	MRL	MRL	90.4	0.8138	0.1059	0.2242	GO
WUN_10	169.1173	-3	-6	MRL	MRL	MRL	9.11	0.0568	0.0002	0.354	GO
WUN_11	104.8861	-2	-5	MRL	MRL	MRL	1827	0.1128	0.0004	1.69	GO
WUN_12	165.1953	-3	-5	MRL	MRL	MRL	MRL	0.1613	0.0003	2.92	0.9049
WUN_13	178.2629	-2	-3	MRL	MRL	MRL	2198	0.3779	0.0143	0.8053	GO
WUN_14	329.5698	-2	-2	MRL	MRL	MRL	MRL	1.39	0.8911	3.38	0.5581
WUN_15	361.5177	-2	-2	MRL	MRL	MRL	2290	1.20	0.1703	2.73	GO
WUN_16	285.9343	-2	-5	MRL	MRL	MRL	10.1	0.5105	0.0004	0.0028	GO
WUN_17	157.0944	-1	-1	3.2	3.05	MRL	MRL	GO	GO	1.53	3.04
WUN_18	238.7333	-3	-4	MRL	MRL	MRL	MRL	0.0074	0.0007	1.30	0.935
WUN_20	403.196	-2	-2	MRL	MRL	MRL	MRL	1.27	0.1212	3.69	3.15
WTN_2	130.7025	-3	-3	17.77	1.35	0.42	0.24	GO	GO	GO	GO
WTN_3	99.49495	-1	-1	0.26	0.26	0.48	0.27	GO	GO	GO	GO
WTN_4	89.83607	-1	-1	0.61	0.4	0.31	0.26	GO	GO	GO	GO
WTN_5	229.7008	-3	-5	MRL	31.05	2.74	2.14	4.05	GO	GO	GO
WTN_6	173.4784	-3	-5	MRL	29.32	13.3	1.73	0.1369	GO	GO	GO
WTN_7	80.77892	-3	-5	MRL	25.27	282	2.55	0.0254	GO	GO	GO

Table 4 continued

Problem	Optimum	Accuracy p		TCPU's			Optimality gap (%)					
		PCM	NMDT	PCM	NMDT	BARON	BARON	NMDT	PCM	BARON	GioMIQO	GioMIQO
WTN_8	109,4014	-2	-4	MRL	MRL	277	MRL	12.9	1.13	GO	GO	GO
WTN_9	124,4421	-1	-2	MRL	MRL	MRL	MRL	MRL	10.1	72.3	6.93	6.93
WTN_10	586.68	-3	-5	MRL	8.22	MRL	8.22	9.83	0.0122	GO	0.2846	GO
WTN_11	2127.115	-1	-1	0.76	0.82	MRL	0.82	9.8	GO	GO	40.2	GO
WTN_12	1201.038	-1	-1	0.85	0.8	MRL	0.8	MRL	GO	GO	34.7	2.11
WTN_13	1564.958	-1	-1	0.75	1.27	MRL	1.27	1178	GO	GO	23.6	GO
WTN_14	513.0009	-3	-6	MRL	203.11	MRL	203.11	MRL	0.1087	GO	26.9	9.79
WTN_15	2446.429	-3	-6	MRL	37.71	303	37.71	2.34	0.112	GO	GO	GO
WTN_16	1358.663	-3	-5	MRL	29.05	MRL	29.05	13.6	0.0697	GO	10.4	GO
HYD_2	209,721	-2	-2	MRL	MRL	MRL	MRL	MRL	0.199	0.199	0.2102	0.4544
HYD_4	371,811.8	-1	-1	MRL	MRL	MRL	MRL	MRL	0.571	1.21	1.21	1.27
HYD_7	744,963.7	-1	-1	6000 ^a	6495 ^a	MRL	6495 ^a	MRL	1.35	2.25	2.25	1.69
MPBP_852	53.9627	-1	-1	12.8	10.9	1090	10.9	MRL	GO	GO	GO	0.1686
MPBP_721	13.5268	-1	-1	7.04	7.07	728	7.07	232	GO	GO	GO	GO
MPBP_718	7.3936	-2	-2	MRL	85.3	1330	85.3	MRL	22.1	GO	GO	41.4
MPBP_531	20.039	-2	-2	MRL	41.46	210	41.46	MRL	1.48	GO	GO	3.14
MPBP_480	9.2266	-2	-2	MRL	MRL	493	MRL	MRL	3.57	GO	GO	0.9148
MPBP_146	45.29674	-2	-4	MRL	MRL	1019	MRL	MRL	0.1554	GO	GO	2.21
MPBP_029	13.3594	-1	-1	0.94	0.78	2.95	0.78	5.52	GO	GO	GO	GO

MRL maximum resource limit
 GO global optimal solution (optimality gap ≤ 0.0001 %)
^a Out of memory termination

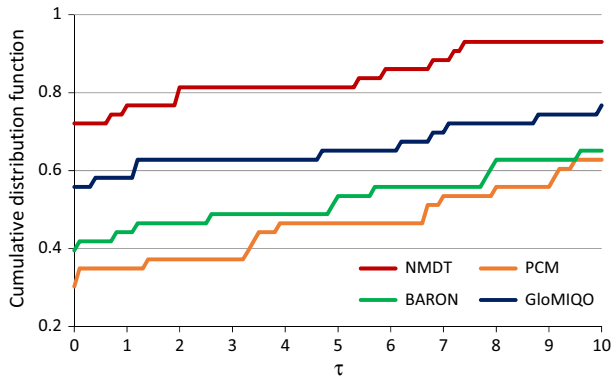


Fig. 6 Performance profile for optimality gap

with the help of the performance profiles of Dolan and Moré [48]. The optimality gap is selected as the performance measure, leading to the chart in Fig. 6.

Interestingly, there is a clear ranking amongst the global optimization algorithms tested: $\text{NMDT} > \text{GloMIQO} > \text{BARON} > \text{PCM}$, which remains constant throughout the x-axis. Note that the chosen upper bound ($\tau = 10$) will count problems for which the optimality gap is at most three orders of magnitude the gap returned by the best algorithm. Thus, if an algorithm can solve a problem to global optimality, that problem will only contribute to the cumulative distribution function of the other algorithms if they return a gap below 0.1%. In this case, the probability that a solver is within such region is equal to 93% for NMDT, 76.7% for GloMIQO and 65.1% for BARON, which is an indication that the NMDT and commercial algorithms have complementary strengths (the gap for NMDT was always at least as good as the one for PCM). Overall, NMDT can obtain consistently low optimality gaps, with a 72.1% probability that it will return the lowest gap amongst all algorithms within the given maximum computational time. The highest gap (3.57%) is obtained for MPBP_480, which can actually be reduced to just 0.30% if more time is given (5877 CPUs). In contrast, GloMIQO and BARON return gaps as high as 41.4 and 72.3%.

7 Conclusions

This paper has presented a mixed integer linear programming model for the relaxation of mixed-integer quadratically constrained problems. It can be viewed as an upgrade of the computationally efficient multiparametric disaggregation technique that relies of the concept of discretizing one of the variables in every bilinear term to a certain accuracy level. The novel aspect concerns the discretization of the range between the variable's bounds rather than its actual values, leading to a normalized approach that is more suitable for integration with bound contracting procedures that are part of spatial branch and bound algorithms.

It has been shown that accuracy levels in normalized multiparametric disaggregation can be related to the number of uniform partitions of a piecewise McCormick relaxation approach, thus allowing for a direct comparison. Through the solution of four sets of NLP and MINLP problems from the literature involving strictly bilinear terms, we have seen that the relaxation quality for a given number of partitions is the same. For the lowest possible accuracy setting, the new approach already outperforms the particular piecewise McCormick formulation presented in the paper, becoming significantly faster with the increase in the number of

partitions due to the logarithmic vs. linear increase in problem size. This is translated into an ability to generate optimality gaps that can be orders of magnitude narrower.

The relaxation problem has been used as basis for a rigorous global optimization algorithm that was compared to commercial solvers BARON and GloMIQO. It has been shown that the proposed simple algorithm outperforms its state-of-the-art counterparts in forty-three benchmark problems, clearly indicating that multiparametric disaggregation should be used to a greater extent.

Acknowledgments Financial support from Fundação para a Ciência e Tecnologia (FCT) through the Investigador FCT 2013 program.

References

1. Quesada, I., Grossmann, I.E.: Global optimization of bilinear process networks with multicomponent flows. *Comput. Chem. Eng.* **19**(12), 1219–1242 (1995)
2. Meyer, C.A., Floudas, C.A.: Global optimization of a combinatorially complex generalized pooling problem. *AIChE J.* **52**(2), 1027–1037 (2006)
3. Misener, R., Thompson, J.P., Floudas, C.A.: APOGEE: global optimization of standard, generalized, and extended pooling problems via linear and logarithmic partitioning schemes. *Comput. Chem. Eng.* **35**, 876–892 (2011)
4. Lee, H., Pinto, J.M., Grossmann, I.E., Park, S.: Mixed-integer linear programming model for refinery short-term scheduling of crude oil unloading with inventory management. *Ind. Eng. Chem. Res.* **35**, 1630–1641 (1996)
5. Jia, Z., Ierapetritou, M., Kelly, J.D.: Refinery short-term scheduling using continuous time formulation: crude-oil operations. *Ind. Eng. Chem. Res.* **42**, 3085–3097 (2003)
6. Castro, P.M., Grossmann, I.E.: Global optimal scheduling of crude oil blending operations with RTN continuous-time and multiparametric disaggregation. *Ind. Eng. Chem. Res.* **53**, 15127–15145 (2014)
7. Moro, L.F.L., Zanin, A.C., Pinto, J.M.: A planning model for refinery diesel production. *Comput. Chem. Eng.* **22**, S1039–42 (1998)
8. Jia, Z., Ierapetritou, M.: Mixed-integer linear programming model for gasoline blending and distribution scheduling. *Ind. Eng. Chem. Res.* **42**, 825–835 (2003)
9. Kolodziej, S.P., Grossmann, I.E., Furman, K.C., Sawaya, N.W.: A discretization-based approach for the optimization of the multiperiod blend scheduling problem. *Comput. Chem. Eng.* **53**, 122–142 (2013)
10. Galan, B., Grossmann, I.E.: Optimal design of distributed wastewater treatment networks. *Ind. Eng. Chem. Res.* **37**, 4036–4048 (1998)
11. Karuppiah, R., Grossmann, I.E.: Global optimization for the synthesis of integrated water systems in chemical processes. *Comput. Chem. Eng.* **30**, 650–673 (2006)
12. Faria, D.C., Bagajewicz, M.J.: Novel bound contracting procedure for global optimization of bilinear MINLP problems with applications to water management problems. *Comput. Chem. Eng.* **35**, 446–55 (2011)
13. Faria, D.C., Bagajewicz, M.J.: A new approach for global optimization of a class of MINLP problems with applications to water management and pooling problems. *AIChE J.* **58**(8), 2320–35 (2012)
14. Rubio-Castro, E., Ponce-Ortega, J.M., Serna-González, M., El-Halwagi, M.M., Pham, V.: Global optimization in property-based inter-plant water integration. *AIChE J.* **59**(3), 813–33 (2013)
15. Carrión, M., Arroyo, J.M.: A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. *IEEE Trans. Power Syst.* **21**(3), 1371–1378 (2006)
16. Catalão, J.P.S., Pousinho, H.M.I., Mendes, V.M.F.: Hydro energy systems management in Portugal: profit-based evaluation of a mixed-integer nonlinear approach. *Energy* **36**, 500–507 (2011)
17. Harjunkoski, I., Westerlund, T., Pörn, R., Skrifvars, H.: Different transformations for solving non-convex trim loss problems by MINLP. *Eur. J. Oper. Res.* **105**, 594–603 (1998)
18. Zorn, K., Sahinidis, N.V.: Computational experience with applications of bilinear cutting planes. *Ind. Eng. Chem. Res.* **52**, 7514–7525 (2013)
19. Teles, J.P., Castro, P.M., Matos, H.A.: Univariate parameterization for global optimization of mixed-integer polynomial problems. *Eur. J. Oper. Res.* **229**, 613–25 (2013)
20. Misener, R., Floudas, C.A.: GloMIQO: global mixed-integer quadratic optimizer. *J. Glob. Optim.* **53**, 3–50 (2013)

21. Sahinidis, N.: BARON: a general purpose global optimization software package. *J. Glob. Optim.* **8**, 201–205 (1996)
22. Misener, R., Floudas, C.A.: ANTIGONE: algorithms for continuous/integer global optimization of non-linear equations. *J. Glob. Optim.* **59**, 503–526 (2014)
23. Castro, P.M., Grossmann, I.E.: Optimality-based bound contraction with multiparametric disaggregation for the global optimization of mixed-integer bilinear problems. *J. Glob. Optim.* **59**, 277–306 (2014)
24. Castro, P.M.: Tightening piecewise McCormick relaxations for bilinear problems. *Comput. Chem. Eng.* **72**, 300–311 (2015)
25. Bergamini, M.L., Aguirre, P., Grossmann, I.E.: Logic-based outer approximation for globally optimal synthesis of process networks. *Comput. Chem. Eng.* **29**, 1914–1933 (2005)
26. Gounaris, C.E., Misener, R., Floudas, C.A.: Computational comparison of piecewise-linear relaxations for pooling problems. *Ind. Eng. Chem. Res.* **48**, 5742–5766 (2009)
27. Wicaksono, D.N., Karimi, I.A.: Piecewise MILP under- and overestimators for global optimization of bilinear programs. *AIChE J.* **54**, 991–1008 (2008)
28. Hasan, M.M.F., Karimi, I.A.: Piecewise linear relaxation of bilinear programs using bivariate partitioning. *AIChE J.* **56**, 1880–1893 (2010)
29. Teles, J.P., Castro, P.M., Matos, H.A.: Multiparametric disaggregation technique for global optimization of polynomial programming problems. *J. Glob. Optim.* **55**, 227–251 (2013)
30. Kolodziej, S., Castro, P.M., Grossmann, I.E.: Global optimization of bilinear programs with a multiparametric disaggregation technique. *J. Glob. Optim.* **57**, 1039–1063 (2013)
31. Misener, R., Floudas, C.A.: Global optimization of mixed-integer quadratically-constrained quadratic programs (MIQCQP) through piecewise-linear and edge-concave relaxations. *Math. Program. Ser. B* **136**(1), 155–82 (2012)
32. Castro, P.M., Teles, J.P.: Comparison of global optimization algorithms for the design of water-using networks. *Comput. Chem. Eng.* **52**, 249–261 (2013)
33. Vielma, J.P., Ahmed, S., Nemhauser, G.: Mixed-integer models for nonseparable piecewise linear optimization: unifying framework and extensions. *Oper. Res.* **58**(2), 303–315 (2009)
34. Teles, J.P., Castro, P.M., Matos, H.A.: Global optimization of water networks design using multiparametric disaggregation. *Comput. Chem. Eng.* **40**, 132–147 (2012)
35. Balas, E.: Disjunctive programming. *Ann. Discrete Math.* **5**, 3–51 (1979)
36. Raman, R., Grossmann, I.E.: Modeling and computational techniques for logic based integer programming. *Comput. Chem. Eng.* **18**, 563–78 (1994)
37. Balas, E.: Disjunctive programming and a hierarchy of relaxations for discrete optimization problems. *SIAM J. Algebr. Discrete Math.* **6**, 466–486 (1985)
38. Jeroslow, R.G., Lowe, J.K.: Modelling with integer variables. *Math. Program. Study* **22**, 167–184 (1984)
39. Oral, M., Kettani, O.: A linearization procedure for quadratic and cubic mixed-integer problems. *Oper. Res.* **40**(Suppl 1), S109–S116 (1992)
40. McCormick, G.P.: Computability of global solutions to factorable nonconvex programs. Part I. Convex underestimating problems. *Math. Program.* **10**, 146 (1976)
41. Sherali, H.D., Alameddine, A.: A new reformulation linearization technique for bilinear programming problems. *J. Glob. Optim.* **2**, 379–410 (1992)
42. Sherali, H.D., Adams, W.P., Driscoll, P.J.: Exploiting special structures in constructing a hierarchy of relaxations for 0–1 mixed integer problems. *Oper. Res.* **46**(3), 396–405 (1998)
43. Hock, W., Schittkowski, K.: Test Examples for Nonlinear Programming Codes, vol. 187 of Lecture Notes in Economics and Mathematical Systems. Springer, Berlin (1981)
44. Floudas, C.A., Pardalos, P.M., Adjiman, C.S., Esposito, W.R., Gumus, Z.H., Harding, S.T., Klepeis, J.L., Meyer, C.A., Schweiger, C.A.: Handbook of Test Problems in Local and Global Optimization. Kluwer, Boston (1999)
45. Misener, R., Floudas, C.A.: Mixed-Integer Quadratically-Constrained Quadratic Programs: GloMIQO 2.2 Test Suite. http://helios.princeton.edu/GloMIQO/MisenerFloudas_GloMIQO_TestSet.pdf. Accessed February 17, 2015
46. Drud, A.S.: CONOPT—a large-scale GRG code. *INFORMS J. Comput.* **6**(2), 207–216 (1994)
47. Tawarmalani, M., Sahinidis, N.V.: A polyhedral branch-and-cut approach to global optimization. *Math. Program.* **103**(2), 225–49 (2005)
48. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. *Math. Program. Ser. A* **91**, 201–213 (2002)
49. Castro, P.M., Grossmann, I.E.: Generalized disjunctive programming as a systematic modeling framework to derive scheduling formulations. *Ind. Eng. Chem. Res.* **51**, 5781–5792 (2012)
50. Duran, M.A., Grossmann, I.E.: An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Math. Program.* **36**(3), 307–339 (1986)