Computers and Chemical Engineering 52 (2013) 249–261

Contents lists available at SciVerse ScienceDirect

Computers and Chemical Engineering

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

Comparison of global optimization algorithms for the design of water-using networks

Pedro M. Castro*, João P. Teles

Unidade de Modelação e Optimização de Sistemas Energéticos, Laboratório Nacional de Energia e Geologia, 1649-038 Lisboa, Portugal

ARTICLE INFO

Article history:
Received 26 July 2012
Received in revised form 22 January 2013
Accepted 23 January 2013
Available online 8 February 2013

Keywords:
Mathematical modeling
Nonlinear programming
Integer programming
Disjunctive programming
Process design
Water minimization

ABSTRACT

We address a special class of bilinear process network problems with global optimization algorithms iterating between a lower bound provided by a mixed-integer linear programming (MILP) formulation and an upper bound given by the solution of the original nonlinear problem (NLP) with a local solver. Two conceptually different relaxation approaches are tested, piecewise McCormick envelopes and multiparametric disaggregation, each considered in two variants according to the choice of variables to partition/parameterize. The four complete MILP formulations are derived from disjunctive programming models followed by convex hull reformulations. The results on a set of test problems from the literature show that the algorithm relying on multiparametric disaggregation with parameterization of the concentrations is the best performer, primarily due to a logarithmic as opposed to linear increase in problem size with the number of partitions. The algorithms are also compared to the commercial solvers BARON and GloMIQO through performance profiles.

1. Introduction

Water networks (Bagajewicz, 2000; Jezowski, 2010) are a special type of process network problems (Quesada & Grossmann, 1995) that can be formulated as non-convex bilinear programs. Another example is the pooling problem (Haverly, 1978), which has been receiving considerable attention in the literature (Faria & Bagajewicz, 2012b; Meyer & Floudas, 2006; Misener & Floudas, 2010; Misener, Thompson, and Floudas, 2011). The non-convex bilinear terms arise in the mixing of streams with different properties and are known to give rise to a multiplicity of local optima, which prevent gradient based solvers from certifying optimality of the nonlinear program (NLP). Further complexity will arise from binary decisions associated to alternative process units and/or connecting pipelines, leading to a mixed-integer nonlinear program (MINLP).

The most common global optimization algorithms are based on spatial branch and bound (Ryoo & Sahinidis, 1996) and involve multiple formulations of a lower bounding problem that is a relaxation of the original bilinear problem. The relaxations are frequently based on the standard McCormick (1976) envelopes, involving the variables domain still in consideration (full domain in the root node) and leading to a linear problem (LP), or on piecewise envelopes (Bergamini, Aguirre, & Grossmann, 2005; Karuppiah & Grossmann, 2006; Meyer & Floudas, 2006). In the latter, the domain of the variables is divided ab initio into a given number of partitions, with the purpose of generating multiple McCormick envelopes that will provide a tighter relaxation. The optimal set of partitions is identified through binary variables leading to a mixed-integer linear programming (MILP) problem.

Partitioning of the search space can be done with respect to one or both variables of the bilinear term. Wicaksono and Karimi (2008), Hasan and Karimi (2010), and Faria and Bagajewicz (2012a) have performed extensive computational tests for a variety of schemes, the former deriving the alternative MILP formulations from a disjunctive program (Balas, 1979). Between the two most common reformulation methods (Balas, 1985), the convex hull approach was found to be superior to its big-M counterpart. More importantly, these schemes require a number of binary variables proportional to the number of partitions, which given the fact that many partitions may be required to achieve a good relaxation, i.e. a low optimality gap, may result in a prohibitively large MILP. To overcome this limitation, Vielma, Ahmed, and Nemhauser (2010), Vielma and Nemhauser (2011), and Misener, Thompson, and Floudas (2011), have proposed MILP formulations that use a logarithmic number of binary variables.

An alternative global optimization approach called multiparametric disaggregation has recently been proposed by Teles, Castro, and Matos (2013) for polynomial problems, which shares the important property of using a logarithmic number of binary variables. Through the discretization of the domain of one of the variables of the bilinear term, the original problem can be

* Corresponding author. Tel.: +351 210924643.
E-mail address: pedro.castro@lneg.pt (P.M. Castro).

0098-1354/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compchemeng.2013.01.013

approximated to any desired accuracy by an MILP. The parameterization (discretization) is based on the concept of significant digits and can be performed with respect to different numeric representation systems (e.g. decimal, binary). One major issue is that the approximation MILP can be infeasible even if the original NLP is feasible, an outcome more likely to occur for lower accuracy levels. Furthermore, the global optimal solution can only be found in the limit of an infinite number of significant digits, which is hardly practical. When tested on the water-using networks (WUN) design problems that are the subject of this work (Teles, Castro, & Matos, 2012a), there were a few examples for which no feasible solution of the MILP could be found in reasonable time and others for which the subsequent (minor) improvement with a local solver, failed to generate the global optimal solution.

To overcome the absence of a lower bounding scheme, Teles, Castro, and Matos (2012b) proposed a rigorous relaxation approach for mixed-integer polynomial problems that effectively makes the domain of parameterized variables continuous by adding residual variables and new sets of constraints to the MILP formulation resulting from multiparametric disaggregation. The practical advantages are the guarantee of feasibility for the MILP, provided that the original MINLP is feasible, and the possibility of computing the optimality gap. This when taking into consideration the upper bound from the solution of the MINLP with a fast local optimization solver. The ability of becoming increasingly tight with improved accuracy is kept, providing the natural iterations of a novel optimization algorithm that is going to be explored in this paper using WUN design problems as case study. Teles et al. (2012b) have also shown that the choice of parameterized variables can have a significant effect on performance, and so we consider the alternative of parameterizing the flowrate variables in addition to that of parameterizing the concentrations already explored in Teles et al. (2012a) with the upper bounding approach.

The main novelty of this paper is to show that multiparametric disaggregation is conceptually different from piecewise McCormick. Instead of comparing the complete MILP formulations, we focus on the essential constraints that can be identified by finding their corresponding linear disjunctive programming models. In other words, we will be deriving the multiparametric disaggregation MILP constraints from the convex hull relaxation of the linear disjunctive program to demonstrate that the two relaxation strategies are different. This strategy of isolating the two modeling decisions (underlying concept and reformulation approach) has provided valuable insights in the case of scheduling problems (Castro & Grossmann, 2012).

The main issue of the MILP relaxation problems derived from both multiparametric disaggregation and piecewise McCormick envelopes is that in general, convergence to the global optimal solution of the original problem can only be guaranteed for an infinite number of discretization points (Kolodziej, Castro, & Grossmann, 2012). Since the optimality gap may be closed much sooner, it is far more efficient to start with a low accuracy level (significantly fewer binary variables), solve the problem, measure the gap, and keep solving the problem following an increase in the accuracy level (extra binary variables and longer computational time), until the gap is below the chosen tolerance or the maximum time limit is exceeded. This is the essence of the newly proposed global optimization algorithm.

The rest of the paper is structured as follows. The problem statement for the WUN design problem and NLP formulation are given in Section 2. In Section 3, the NLP model is reformulated to reduce the number of nonlinear constraints and bilinear terms. The MILP relaxations using piecewise McCormick are the subject of Section 4, where the alternatives of parameterizing the flowrates or concentrations are explored. Section 5 features the relaxation formulations for multiparametric disaggregation, while Section 6 gives the specifics for the WUN design problem. Section 7 discusses the required adjustments when in the presence of a MINLP instead of a NLP. The global optimization algorithm is then the subject of Section 8, with Section 9 being concerned with the analysis of the computational results. Finally, the conclusions can be found in Section 10.

2. Problem statement

We consider the following water-using networks design problem. Given a set $w \in W$ of freshwater sources containing a number of pollutants $c \in C$ with known concentrations ($c^{W}_{w,c}$), the objective is to minimize total freshwater intake while satisfying the demands of a set $i \in I$ of water-using units. According to Prakash and Shenoy (2005), water-using units can be divided in two subsets: (a) fixed load operations I^{fl}, characterized by a limiting flowrate ($f^{U-\lim}_{i}$) and maximum inlet and outlet concentrations ($\bar{c}^{U-in}_{i,c}$ and $\bar{c}^{U-out}_{i,c}$); note that the inlet flowrate F^{U}_{i} is a model variable and is assumed to be equal to the outlet flowrate; (b) fixed flowrate operations I^{ff}, defined by inlet (f^{U-in}_{i}) and outlet (f^{U-out}_{i}) flowrates, which can be different; the inlet concentrations can vary up to the maximum limits $\bar{c}^{U-in}_{i,c}$, while it is assumed that the outlet concentrations are fixed ($\bar{c}^{U-out}_{i,c}$) and thus independent of the inlet concentrations.

2.1. NLP formulation

The design of water-using networks can be formulated as a nonlinear program (Teles, Castro, & Novais, 2008, 2009) starting from a superstructure including all design alternatives, where freshwater sources, water-using units and wastewater sink (represented as rectangles), mixers (diamonds) and splitters (circles) are the main elements, see Fig. 1. The flowrate of freshwater source w can be split (SP_w) into as many streams as the number of units (in order to avoid aggregated flowrate variables, these splitters are located outside the system boundary). Freshwater streams can be mixed at MX_i with the outlet streams from the water-using units before going into unit i. In the diagram, Unit 1 represents a fixed load operation, belonging to subset (a), while Unit 2 corresponds to a fixed flowrate operation, subset (b). The outlet stream can then be sent to the wastewater treatment system, reused in other units or recycled. This decision is taken at splitter SP_i.

In the following constraints, and to make it easier to identify bilinear terms, variables are represented in upper case. If one disregards the superscripts and subscripts, F represents a flowrate and C a concentration variable.

Eq. (1) defines the objective function. The flowrate balances over the unit's inlet mixers (MX_i) and outlet splitters (SP_i) are given in Eqs. (2) and (3). The mass balance over the inlet mixers is required for fixed load units, Eq. (4), while for the fixed flowrate units it is enough to ensure that the maximum inlet concentration is not exceeded, Eq. (5). The mass balance over fixed load units is the subject of Eq. (6), where the mass to be removed is calculated from problem data, Eq. (7). The bounds for the concentration variables ($C^{U-in}_{i,c}$ and $C^{U-out}_{i,c}$), Eqs. (8) and (9), complete the model. Note that Eqs. (4)–(6) are nonlinear.

$$\min \ \mathrm{OBJ} = \sum_{w \in W} \sum_{i \in I} F^{W-U}_{w,i} \tag{1}$$

$$F^{U}_{i}\big|_{i \in I^{fl}} + f^{U-in}_{i}\big|_{i \in I^{ff}} = \sum_{w \in W} F^{W-U}_{w,i} + \sum_{i' \in I} F^{U-U}_{i',i} \quad \forall i \in I \tag{2}$$

$$F^{U}_{i}\big|_{i \in I^{fl}} + f^{U-out}_{i}\big|_{i \in I^{ff}} = F^{U-D}_{i} + \sum_{i' \in I} F^{U-U}_{i',i} \quad \forall i \in I \tag{3}$$

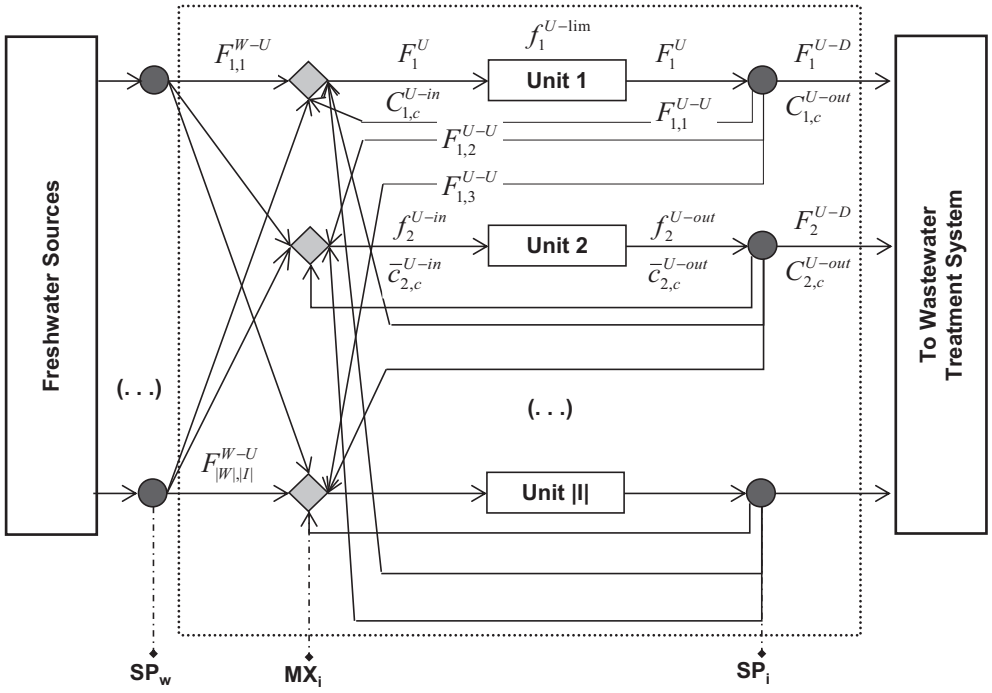

Fig. 1. Superstructure of water-using network.

$$F_i^U \cdot C_{i,c}^{U-in} = \sum_{w \in W} F_{w,i}^{W-U} \cdot c_{w,c}^{W} + \sum_{i' \in I} F_{i',i}^{U-U} \cdot \left(C_{i',c}^{U-out}\Big|_{i' \in I^{fl}} + \bar{c}_{i',c}^{U-out}\Big|_{i' \in I^{ff}} \right) \quad \forall i \in I^{fl}, \quad c \in C \tag{4}$$

$$f_i^{U-in} \cdot \bar{c}_{i,c}^{U-in} \geq \sum_{w \in W} F_{w,i}^{W-U} \cdot c_{w,c}^{W} + \sum_{i' \in I} F_{i',i}^{U-U} \cdot \left(C_{i',c}^{U-out}\Big|_{i' \in I^{fl}} + \bar{c}_{i',c}^{U-out}\Big|_{i' \in I^{ff}} \right) \quad \forall i \in I^{ff}, \quad c \in C \tag{5}$$

$$\Delta m_{i,c}^U = F_i^U \cdot \left(C_{i,c}^{U-out} - C_{i,c}^{U-in} \right) \quad \forall i \in I^{fl}, \quad c \in C \tag{6}$$

$$\Delta m_{i,c}^U = f_i^{U-\lim} \cdot \left(\bar{c}_{i,c}^{U-out} - \bar{c}_{i,c}^{U-in} \right) \quad \forall i \in I^{fl}, \quad c \in C \tag{7}$$

$$C_{i,c}^{U-in} \leq \bar{c}_{i,c}^{U-in} \quad \forall i \in I^{fl}, \quad c \in C \tag{8}$$

$$C_{i,c}^{U-out} \leq \bar{c}_{i,c}^{U-out} \quad \forall i \in I^{fl}, \quad c \in C \tag{9}$$

3. Reformulation linearization of WUN design problem

Previous research (Teles et al., 2008, 2009) has shown that the design problem can be formulated as an LP if the outlet concentrations are parameters instead of variables (e.g. the problem is linear if it consists of only fixed flowrate units). While this is only a valid assumption for single contaminant systems (for which pinch technology can be applied to find the global optimal solution) and is thus not used in this work, it provides valuable insight about the system that can be used to derive a formulation with fewer bilinear terms. Let us focus on the outlet streams from the water-using unit's splitters (SP_i) and define new continuous variables $Z_{i,i',c}^{U-U}$ and $Z_{i,c}^{U-D}$ that obey Eqs. (10) and (11).

$$Z_{i,i',c}^{U-U} = F_{i,i'}^{U-U} \cdot C_{i,c}^{U-out} \tag{10}$$

$$Z_{i,c}^{U-D} = F_i^{U-D} \cdot C_{i,c}^{U-out} \tag{11}$$

If, in addition, variables $M_{i,c}^{U-in}$ and $M_{i,c}^{U-out}$ are defined, representing respectively the inlet and outlet mass of contaminant c in unit i, new sets of linear constraints can be written that replace the original bilinear constraints. More specifically, the mass balances over MX_i, mixers, fixed load units and maximum inlet concentration constraints in linear Eqs. (12)–(14) replace Eqs. (4)–(6), (8). Then, Eq. (15) represents the mass balance over the SP_i splitters. Notice that the resulting NLP formulation Eqs. (1)–(3), (7), (9)–(15) does not include inlet concentrations as model variables.

$$M_{i,c}^{U-in} = \sum_{w \in W} F_{w,i}^{W-U} \cdot c_{w,c}^{W} + \sum_{i' \in I} \left[Z_{i',i,c}^{U-U}\Big|_{i' \in I^{fl}} + \left(F_{i',i}^{U-U} \cdot \bar{c}_{i',c}^{U-out} \right)\Big|_{i' \in I^{ff}} \right] \quad \forall i \in I, \quad c \in C \tag{12}$$

$$\Delta m_{i,c}^U = M_{i,c}^{U-out} - M_{i,c}^{U-in} \quad \forall i \in I^{fl}, \quad c \in C \tag{13}$$

$$\left(F_i^U\Big|_{i \in I^{fl}} + f_i^{U-in}\Big|_{i \in I^{ff}} \right) \cdot \bar{c}_{i,c}^{U-in} \geq M_{i,c}^{U-in} \quad \forall i \in I, \quad c \in C \tag{14}$$

$$M_{i,c}^{U-out} = \sum_{i' \in I} Z_{i,i',c}^{U-U} + Z_{i,c}^{U-D} \quad \forall i \in I^{fl}, \quad c \in C \tag{15}$$

Overall, see Fig. 2, we have used the general ideas of the reformulation linearization technique (RLT) first proposed by Sherali and Alameddine (1992), which can potentially lead to a tighter linear relaxation (Liberti & Pantelides, 2006; Ruiz & Grossmann, 2011) and hence better computational performance. While it is perfectly possible to apply the relaxation approaches over what would be bilinear terms linked to variables $Z_{i,i',c}^{U-U}$, $Z_{i,c}^{U-in}$ and $Z_{i,c}^{U-out}$, the above

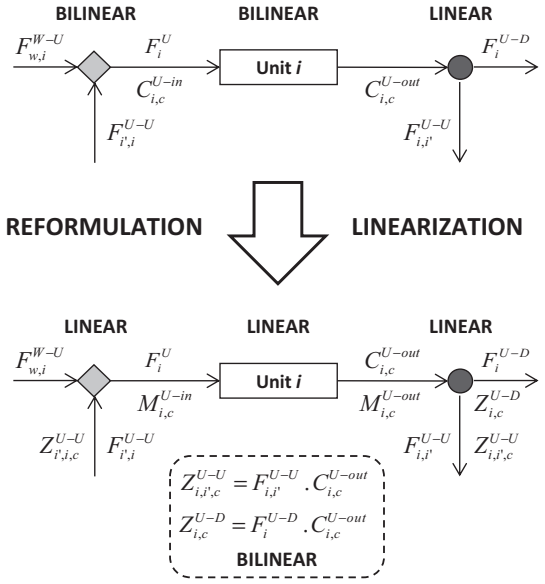

Fig. 2. Reducing the number of bilinear terms in the WUN design problem.

formulation uses one less bilinear term. More specifically, we have removed from the formulation, bilinear terms $F_i^U \cdot C_{i,c}^{U-in}$.

3.1. Remarks

It should be noted at this point that the integrated model proposed by Karuppiah and Grossmann (2006) assumes that the flowrate going through fixed load units is known a priori rather than being an optimization variable. Under such assumption, the only source of bilinear terms involves the mass balance over the MX_i mixers whereas in the current problem definition they also appear in the balances over the units.

4. MILP relaxation of bilinear programs

The bilinear terms in the NLP formulation are frequent in problems from process networks. They are either of type $F_i \cdot C_{i,c}$ or $F_{i,i'} \cdot C_{i,c}$, where F_i ($F_{i,i'}$) represents the flowrate associated to unit i (from unit i to i') and $C_{i,c}$ the corresponding concentration for contaminant c. To make the notation simpler, and since it is straightforward to add index i' to the relevant variables and to the domain of the resulting constraints, we proceed solely with the former term.

McCormick (1976) envelopes in Eq. (16) generate the tightest linear relaxation of bilinear term $Z_{i,c} = F_i \cdot C_{i,c}$, where $\underline{f}_i$ and $\bar{f}_i$ represent the lower and upper bounds of flowrate variable F_i, while $\underline{c}_{i,c}$ and $\bar{c}_{i,c}$ give the bounds for concentration variable $C_{i,c}$. The solution of the LP relaxation may, however, yield a weak lower bound, which is detrimental to the performance of spatial branch and bound methods that rely on this relaxation and that are part of most deterministic global optimization solvers (Sahinidis, 1996).

$$\begin{aligned}
&Z_{i,c} \geq F_i \cdot \underline{c}_{i,c} + \underline{f}_i \cdot C_{i,c} - \underline{f}_i \cdot \underline{c}_{i,c} \quad \forall i, c\\
&Z_{i,c} \geq F_i \cdot \bar{c}_{i,c} + \bar{f}_i \cdot C_{i,c} - \bar{f}_i \cdot \bar{c}_{i,c} \quad \forall i, c\\
&Z_{i,c} \leq F_i \cdot \underline{c}_{i,c} + \bar{f}_i \cdot C_{i,c} - \bar{f}_i \cdot \underline{c}_{i,c} \quad \forall i, c\\
&Z_{i,c} \leq F_i \cdot \bar{c}_{i,c} + \underline{f}_i \cdot C_{i,c} - \underline{f}_i \cdot \bar{c}_{i,c} \quad \forall i, c
\end{aligned} \tag{16}$$

4.1. Piecewise McCormick envelopes (PCM)

In order to strengthen the lower bound resulting from the relaxation, one can use better bounds for the $\underline{f}_i$, $\bar{f}_i$, $\underline{c}_{i,c}$ and $\bar{c}_{i,c}$ parameters in Eq. (16) by partitioning the domain of the variables into n disjoint regions. The so-called piecewise linear McCormick envelopes were first proposed by Bergamini et al. (2005) for a uniform partitioning scheme, which is likely to produce the tightest relaxation (Hasan & Karimi, 2010), and employed among others by Karuppiah and Grossmann (2006) and Meyer and Floudas (2006). Increasing the number of partitions will tighten the relaxation at the expense of a larger and more difficult to solve problem so a trade-off is involved. With the aim of improving efficiency and reducing problem size, more recent work has been focusing on nonuniform (Gounaris, Misener, & Floudas, 2009; Wicaksono & Karimi, 2008), with growing emphasis on logarithmic partitions (Misener et al., 2011).

We now consider two univariate partitioning schemes, starting with the flowrate variables. First, the partitioning scheme is described and the relaxation constraints given in disjunctive program (DP) format (Balas, 1979) to make it easier to compare the different alternatives. Then, we apply the convex hull reformulation (Balas, 1985) together with algebraic manipulations to reduce the number of disaggregated variables and derive a compact formulation of the relaxation in mixed-integer linear format.

4.1.1. Partitioning the flowrate variables (PCM_F)

Karuppiah and Grossmann (2006) and Nápoles-Rivera, Ponce-Ortega, El-Halwagi, and Jiménez-Gutiérrez (2010) used partitioning of the original domain $[\underline{f}_i \quad \bar{f}_i]$ of each flowrate variable F_i appearing in the nonconvex terms, into n uniform intervals with domain $[\underline{f}_{i,n} \quad \bar{f}_{i,n}]$, see Fig. 3. Clearly, it can be assumed that any given value of F_i belongs to a single partition of the domain (in the case of boundary points, one of the two partitions must be selected), i.e. being $Y_{i,n}^F$ Boolean variables indicating if the variable associated to unit i belongs or not to slot n, it is $Y_{i,1}^F = True$, $Y_{i,2}^F = True$, ..., or $Y_{i,|N|}^F = True$. Notice the use of the exclusive OR ($\veebar$) separating the disjunctions.

The next step is to write the McCormick envelopes for each partition, taking advantage of the tighter bounds for the flowrate variables. In contrast, the bounds for the concentration variables $C_{i,c}$ remain the same. The disjunctive programming model for the linear relaxation of bilinear term $Z_{i,c}$ is given in Eq. (17).

$$\begin{aligned}
&\veebar_n \left[\begin{array}{c} Y_{i,n}^F \\
Z_{i,c} \geq F_i \cdot \underline{c}_{i,c} + \underline{f}_{i,n} \cdot C_{i,c} - \underline{f}_{i,n} \cdot \underline{c}_{i,c} \quad \forall c \\
Z_{i,c} \geq F_i \cdot \bar{c}_{i,c} + \bar{f}_{i,n} \cdot C_{i,c} - \bar{f}_{i,n} \cdot \bar{c}_{i,c} \quad \forall c \\
Z_{i,c} \leq F_i \cdot \underline{c}_{i,c} + \bar{f}_{i,n} \cdot C_{i,c} - \bar{f}_{i,n} \cdot \underline{c}_{i,c} \quad \forall c \\
Z_{i,c} \leq F_i \cdot \bar{c}_{i,c} + \underline{f}_{i,n} \cdot C_{i,c} - \underline{f}_{i,n} \cdot \bar{c}_{i,c} \quad \forall c \\
\underline{f}_{i,n} \leq F_i \leq \bar{f}_{i,n} \\
\underline{c}_{i,c} \leq C_{i,c} \leq \bar{c}_{i,c} \quad \forall i, c \\
Y_{i,n}^F = \{True, False\} \quad \forall i, n \end{array}\right] \quad \forall i
\end{aligned} \tag{17}$$

Of the two most common options to reformulate the DP model into a MILP, Karuppiah and Grossmann (2006) used the tighter convex hull relaxation (Balas, 1985). After defining new sets of disaggregated variables (e.g. $\widehat{F}_{i,n}$, $\widehat{C}_{i,c,n}$) for the three sets of variables appearing in the constraints inside the disjunctions, and eliminating those associated to $Z_{i,c}$ by summing the resulting constraints over n, we get the piecewise McCormick envelope in

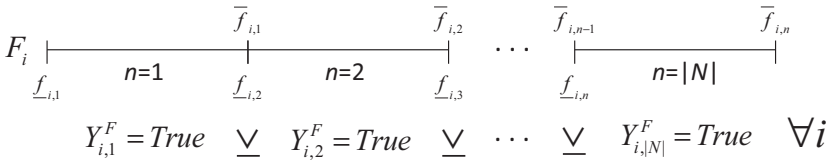

Fig. 3. Partitioning flowrate variables for piecewise McCormick envelopes.

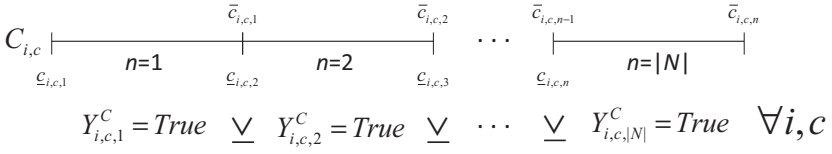

Fig. 4. Partitioning concentration variables for piecewise McCormick envelopes.

Eq. (18). Variables $Y^F_{i,n}$ are now of the binary type (e.g. $Y^F_{i,n} = True$ is equivalent to $Y^F_{i,n} = 1$).

$$\begin{aligned}
Z_{i,c} &\geq \sum_n \left(\widehat{F}_{i,n}\cdot \underline{c}_{i,c} + \underline{f}_{i,n}\cdot \widehat{C}_{i,c,n} - \underline{f}_{i,n}\cdot \underline{c}_{i,c}\cdot Y^F_{i,n}\right) \quad \forall i,c \\
Z_{i,c} &\geq \sum_n \left(\widehat{F}_{i,n}\cdot \bar{c}_{i,c} + \overline{f}_{i,n}\cdot \widehat{C}_{i,c,n} - \overline{f}_{i,n}\cdot \bar{c}_{i,c}\cdot Y^F_{i,n}\right) \quad \forall i,c \\
Z_{i,c} &\leq \sum_n \left(\widehat{F}_{i,n}\cdot \underline{c}_{i,c} + \overline{f}_{i,n}\cdot \widehat{C}_{i,c,n} - \overline{f}_{i,n}\cdot \underline{c}_{i,c}\cdot Y^F_{i,n}\right) \quad \forall i,c \\
Z_{i,c} &\leq \sum_n \left(\widehat{F}_{i,n}\cdot \bar{c}_{i,c} + \underline{f}_{i,n}\cdot \widehat{C}_{i,c,n} - \underline{f}_{i,n}\cdot \bar{c}_{i,c}\cdot Y^F_{i,n}\right) \quad \forall i,c
\end{aligned} \tag{18}$$

The remaining constraints of the convex hull reformulation involve: (i) the relations between the original and disaggregated variables, Eqs. (19) and (20); (ii) the bounding constraints for the disaggregated variables that ensure they are only non-negative for the active partition, Eqs. (21) and (22); (iii) the linear equality constraint Eq. (23) representing the exclusive OR constraint; and (iv) the domain of the new binary and continuous variables, Eq. (24). Notice that the bounds for the concentration variables in Eq. (17) are no longer needed since they are a combination of Eqs. (20), (22), (23).

$$F_i = \sum_n \widehat{F}_{i,n} \quad \forall i \tag{19}$$

$$C_{i,c} = \sum_n \widehat{C}_{i,c,n} \quad \forall i,c \tag{20}$$

$$\underline{f}_{i,n}\cdot Y^F_{i,n} \leq \widehat{F}_{i,n} \leq \overline{f}_{i,n}\cdot Y^F_{i,n} \quad \forall i,n \tag{21}$$

$$\underline{c}_{i,c}\cdot Y^F_{i,n} \leq \widehat{C}_{i,c,n} \leq \bar{c}_{i,c}\cdot Y^F_{i,n} \quad \forall i,c,n \tag{22}$$

$$\sum_n Y^F_{i,n} = 1 \quad \forall i \tag{23}$$

$$Y^F_{i,n} = \{0,1\} \quad \forall i,n \quad \widehat{F}_{i,n} \geq 0 \quad \forall i,n \quad \widehat{C}_{i,c,n} \geq 0 \quad \forall i,c,n \tag{24}$$

4.1.2. Partitioning the concentration variables (PCM_C)

We now explore the alternative of selecting the concentration variables for partitioning. Given that concentrations are one of the properties in the property integration framework of El-Halwagi, Glaswog, Qin, and Eden (2004), this has been the approach of Rubio-Castro, Ponce-Ortega, Serna-González, El-Halwagi, and Pham (2012). For every unit i and contaminant c, the original domain $[\ \underline{c}_{i,c},\ \ \bar{c}_{i,c}\ ]$ is divided into the same number n of uniform slots with domain $[\ \underline{c}_{i,c,n},\ \ \bar{c}_{i,c,n}\ ]$, see Fig. 4. Again, a single partition can be selected for a particular variable, with binary variables $Y^C_{i,c,n}$ identifying the active ones. The corresponding disjunctive programming model for the relaxation of the bilinear term $Z_{i,c}$ is given in Eq. (25), featuring tighter bounds for the concentration variables on each disjunction.

Using the convex hull reformulation followed by aggregation of constraints to remove the disaggregated variables linked to original variables $Z_{i,c}$, leads us to Eqs. (20), (26–31). Despite the obvious differences in the parameters linked to the variables upper and lower bounds, the final result is quite similar to the one for partitioning over the flowrate variables. It should however be highlighted that more variables and constraints are being used. More specifically, the binary variables together with the disaggregated flowrate variables feature the contaminant index c and so do the constraints (compare Eqs. (27) and (28) with Eqs. (19) and (21)). This is a direct consequence of index c being in the domain of the disjunctive set in Eq. (25), whereas in Eq. (17) it is an index of the constraints associated to each disjunction.

$$\begin{aligned}
&\veebar_n \left[\begin{array}{c}
Y^C_{i,c,n} \\
Z_{i,c} \geq F_i\cdot \underline{c}_{i,c,n} + \underline{f}_i\cdot C_{i,c} - \underline{f}_i\cdot \underline{c}_{i,c,n} \\
Z_{i,c} \geq F_i\cdot \bar{c}_{i,c,n} + \overline{f}_i\cdot C_{i,c} - \overline{f}_i\cdot \bar{c}_{i,c,n} \\
Z_{i,c} \leq F_i\cdot \underline{c}_{i,c,n} + \overline{f}_i\cdot C_{i,c} - \overline{f}_i\cdot \underline{c}_{i,c,n} \\
Z_{i,c} \leq F_i\cdot \bar{c}_{i,c,n} + \underline{f}_i\cdot C_{i,c} - \underline{f}_i\cdot \bar{c}_{i,c,n} \\
\underline{c}_{i,c,n} \leq C_{i,c} \leq \bar{c}_{i,c,n} \\
\underline{f}_i \leq F_i \leq \overline{f}_i \quad \forall i \\
Y^C_{i,c,n} = \{True, False\} \quad \forall i,c,n
\end{array}\right] \quad \forall i,c \\
&Z_{i,c} \geq \sum_n \left(\widehat{F}_{i,c,n}\cdot \underline{c}_{i,c,n} + \underline{f}_i\cdot \widehat{C}_{i,c,n} - \underline{f}_i\cdot \underline{c}_{i,c,n}\cdot Y^C_{i,c,n}\right) \quad \forall i,c \\
&Z_{i,c} \geq \sum_n \left(\widehat{F}_{i,c,n}\cdot \bar{c}_{i,c,n} + \overline{f}_i\cdot \widehat{C}_{i,c,n} - \overline{f}_i\cdot \bar{c}_{i,c,n}\cdot Y^C_{i,c,n}\right) \quad \forall i,c \\
&Z_{i,c} \leq \sum_n \left(\widehat{F}_{i,c,n}\cdot \underline{c}_{i,c,n} + \overline{f}_i\cdot \widehat{C}_{i,c,n} - \overline{f}_i\cdot \underline{c}_{i,c,n}\cdot Y^C_{i,c,n}\right) \quad \forall i,c \\
&Z_{i,c} \leq \sum_n \left(\widehat{F}_{i,c,n}\cdot \bar{c}_{i,c,n} + \underline{f}_i\cdot \widehat{C}_{i,c,n} - \underline{f}_i\cdot \bar{c}_{i,c,n}\cdot Y^C_{i,c,n}\right) \quad \forall i,c
\end{aligned} \tag{25, 26}$$

$$F_i = \sum_n \widehat{F}_{i,c,n} \quad \forall i,c \tag{27}$$

$$\underline{f}_i\cdot Y^C_{i,c,n} \leq \widehat{F}_{i,c,n} \leq \overline{f}_i\cdot Y^C_{i,c,n} \quad \forall i,c,n \tag{28}$$

$$\underline{c}_{i,c,n}\cdot Y^C_{i,c,n} \leq \widehat{C}_{i,c,n} \leq \bar{c}_{i,c,n}\cdot Y^C_{i,c,n} \quad \forall i,c,n \tag{29}$$

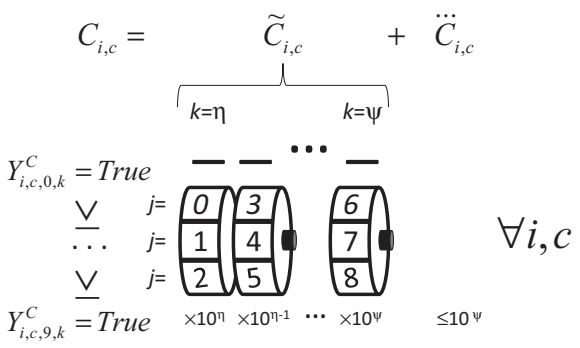

Fig. 5. Parameterization of concentration variables for multiparametric disaggregation.

$$\sum_n Y^C_{i,c,n} = 1 \quad \forall i, c \tag{30}$$

$$Y^C_{i,c,n} = \{0, 1\} \quad \forall i, c, n \quad \widehat{F}_{i,c,n} \geq 0 \quad \forall i, c, n \quad \widehat{C}_{i,c,n} \geq 0 \quad \forall i, c, n \tag{31}$$

Since a larger size is typically associated to a worse computational performance, it may seem awkward to consider the partition route over the concentration variables. Recall, however, that splitter nodes (SP_i) have multiple output streams sharing the same concentration. Thus, partitioning over flowrate variables with two unit indices may lead to larger problems than over concentration variables.

5. MILP relaxation using multiparametric disaggregation (MDT)

In our recently proposed multiparametric disaggregation technique (Teles et al., 2013, 2012a), one of the variables in the bilinear term is parameterized (discretized) rather than partitioned while the other is disaggregated. The method can approximate discretized variables to any desired accuracy level through the use of additional continuous and binary variables, leading to an upper bounding MILP. Different numeric representation systems can be used for the parameterization (Teles et al., 2012a), with the decimal system leading to the simplest nomenclature. This is the primary reason for its selection in the follow up. Nevertheless, we consider the binary system in the computational results section.

We will now derive the linear relaxation for the bilinear term starting from a disjunctive program. Since the upper bounding formulation has already been tested in water-using networks by parameterizing the concentration variables (Teles et al., 2012a), this is the first alternative considered.

5.1. Parameterizing the concentration variables (MDT_C)

The multiparametric disaggregation technique builds on the fact that the concentration variable can be approximated by truncating in two points, η and ψ, with $\eta - \psi + 1$ giving the number of significant digits used in the approximation. While $\eta = \lfloor \log_{10} \bar{c}_{i,c} \rfloor$, the value of ψ is related to the desired accuracy and can belong either to the integral or decimal region. In order to provide a relaxation to the original problem, which generates a lower bound, we now propose to represent the value for the concentration variable $C_{i,c}$ as a sum of approximation ($\tilde{C}_{i,c}$) and residual terms ($\ddot{C}_{i,c}$), see Fig. 5.

For the representation of the approximation term, we do as in Teles et al. (2012a) and consider all significant positions $k \in \{\psi, \ldots, \eta\}$ and all possible digits $j \in \{0, \ldots, 9\}$. The discrete concentration value in the relaxation problem will then be found by multiplying the active digit j by the proper power of ten, 10^k, and summing over all significant positions. As for the residual concentration, it is bounded by the difference between any two consecutive discrete points, 10^{ψ}, to make sure that the full domain is considered. As an example, $C_{i,c} = 125.643$ can be obtained with $\eta = 2$ and $\psi = -1$ by making $\tilde{C}_{i,c} = 1 \cdot 10^2 + 2 \cdot 10^1 + 5 \cdot 10^0 + 6 \cdot 10^{-1}$ and $\ddot{C}_{i,c} = 0.43 \cdot 10^{-1}$.

Similarly, the exact representation of the bilinear term can be divided into an approximate and residual term, Eq. (32). The problem of finding the value of the approximate bilinear term can actually be modeled as a disjunctive program, where Boolean variables $Y^C_{i,c,j,k}$ identify if the concentration of unit i, contaminant c, features digit j in position k, see Eq. (33). It is written for for every unit i, contaminant c and position k.

$$Z_{i,c} = F_i \cdot (\tilde{C}_{i,c} + \ddot{C}_{i,c}) = F_i \cdot \tilde{C}_{i,c} + F_i \cdot \ddot{C}_{i,c} = F_i \cdot \sum_{k=\psi}^{\eta} \tilde{C}_{i,c,k} + F_i \cdot \ddot{C}_{i,c}$$
$$= \sum_{k=\psi}^{\eta} \tilde{Z}_{i,c,k} + \ddot{Z}_{i,c} = \tilde{Z}_{i,c} + \ddot{Z}_{i,c} \tag{32}$$

$$\bigvee_j \begin{bmatrix} Y^C_{i,c,j,k} \\ \tilde{C}_{i,c,k} = j \cdot 10^k \\ \tilde{Z}_{i,c,k} = F_i \cdot j \cdot 10^k \end{bmatrix} \quad \forall i, c, k \in \{\psi, \ldots, \eta\} \tag{33}$$

Applying the convex hull reformulation and aggregating the constraints over j to avoid the need for the disaggregated variables linked to the concentration and bilinear term variables, we get to the MILP model that provides the approximate representation of the values for position k, Eqs. (34) and (35). Notice that the Boolean variables have again been converted to binary variables.

$$\begin{aligned} \tilde{C}_{i,c,k} &= \sum_{j=0}^{9} j \cdot 10^k \cdot Y^C_{i,c,j,k} \quad \forall i, c, k \in \{\psi, \ldots, \eta\} \\ \tilde{Z}_{i,c,k} &= \sum_{j=0}^{9} \widehat{F}_{i,c,j,k} \cdot j \cdot 10^k \quad \forall i, c, k \in \{\psi, \ldots, \eta\} \end{aligned} \tag{34}$$

$$\begin{aligned} F_i &= \sum_{j=0}^{9} \widehat{F}_{i,c,j,k} \quad \forall i, c, k \in \{\psi, \ldots, \eta\} \\ \underline{f}_i \cdot Y^C_{i,c,j,k} \leq \widehat{F}_{i,c,j,k} &\leq \overline{f}_i \cdot Y^C_{i,c,j,k} \quad \forall i, c, j \in \{0, \ldots, 9\}, \; k \in \{\psi, \ldots, \eta\} \\ \sum_{j=0}^{9} Y^C_{i,c,j,k} &= 1 \quad \forall i, c, k \in \{\psi, \ldots, \eta\} \end{aligned} \tag{35}$$

The summation over k of variables $\tilde{C}_{i,c,k}$ and $\tilde{Z}_{i,c,k}$ will give us the approximate values for the concentration and bilinear term variables. The upper bounding problem is thus made up of Eqs. (35) and (36).

$$\begin{aligned} \tilde{C}_{i,c} &= \sum_{k=\psi}^{\eta} \sum_{j=1}^{9} j \cdot 10^k \cdot Y^C_{i,c,j,k} \quad \forall i, c \\ \tilde{Z}_{i,c} &= \sum_{k=\psi}^{\eta} \sum_{j=1}^{9} \widehat{F}_{i,c,j,k} \cdot j \cdot 10^k \quad \forall i, c \end{aligned} \tag{36}$$

In order to generate the lower bounding formulation, the bilinear residual term $\ddot{Z}_{i,c} = F_i \cdot \ddot{C}_{i,c}$ is relaxed using the McCormick

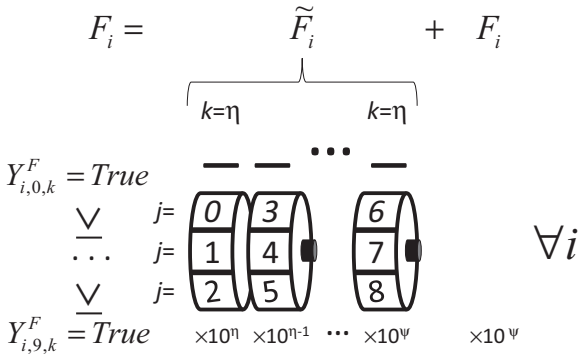

Fig. 6. Parameterization of flowrate variables for multiparametric disaggregation.

(1976) envelopes, see Eq. (37), where the bounds on the residual concentration have been used. The complete lower bounding formulation is given by Eqs. (35), (37)–(42). The global McCormick envelopes in Eq. (16) are also added to improve the linear relaxation of the formulation.

$$\begin{aligned} &\underline{f}_i \cdot \ddot{C}_{i,c} \le \ddot{Z}_{i,c} \le \bar{f}_i \cdot \ddot{C}_{i,c} \quad \forall i, c \\ &(F_i - \bar{f}_i) \cdot 10^{\psi} + \bar{f}_i \cdot \ddot{C}_{i,c} \le \ddot{Z}_{i,c} \le (F_i - \underline{f}_i) \cdot 10^{\psi} + \underline{f}_i \cdot \ddot{C}_{i,c} \quad \forall i, c \end{aligned} \tag{37}$$

$$\begin{aligned} C_{i,c} &= \sum_{k=\psi}^{\eta} \sum_{j=1}^{9} j \cdot 10^k \cdot Y^C_{i,c,j,k} + \ddot{C}_{i,c} \quad \forall i, c \\ Z_{i,c} &= \sum_{k=\psi}^{\eta} \sum_{j=1}^{9} \widehat{F}_{i,c,j,k} \cdot j \cdot 10^k + \ddot{Z}_{i,c} \quad \forall i, c \end{aligned} \tag{38}$$

$$\underline{c}_{i,c} \le C_{i,c} \le \bar{c}_{i,c} \quad \forall i, c \tag{39}$$

$$0 \le \ddot{C}_{i,c} \le 10^{\psi} \quad \forall i, c \tag{40}$$

$$\widehat{F}_{i,c,j,k} \ge 0 \quad \forall i, c, j \in \{0, \ldots, 9\}, \quad k \in \{\psi, \ldots, \eta\} \tag{41}$$

$$Y^C_{i,c,j,k} = \{0, 1\} \quad \forall i, c, j \in \{0, \ldots, 9\}, \quad k \in \{\psi, \ldots, \eta\} \tag{42}$$

When compared to the MILP model resulting from the piecewise McCormick concentrations partitioning approach, see Section 4.1.2, no disaggregated concentration variables are now required. As for the remaining additional binary and continuous variables, the difference is in the domain, which now features two indices, j and k, instead of the partition index n. Note that fewer variables will be required to achieve the same accuracy level beyond one significant digit. More specifically, and assuming a domain between [0,10[, an accuracy equal to 0.1 will require 10·2 = 20 rather than 100 binary variables per concentration variable. For an accuracy of 0.001 we already need 10,000 partitions for the piecewise McCormick approach, a much larger number than the 40 binary variables needed for the multiparametric disaggregation method.

5.2. Parameterization of flowrate variables (MDT_F)

For completeness, we also consider the alternative of parameterizing the flowrate instead of the concentration variables. Assuming that the exact value for the flowrate variable of operation i, F_i, is given by the sum of approximated $\tilde{F}_i$ and residual values $\ddot{F}_i$, Fig. 6 illustrates the parameterization part of the method. The disjunctive programming model for the approximate representation is then given in Eq. (43), where index c is now associated to the constraints inside a particular disjunction and not to the set of disjunctions, compare to Eq. (33).

$$\bigvee_j \begin{bmatrix} Y^F_{i,j,k} \\ \tilde{F}_{i,k} = j \cdot 10^k \\ \tilde{Z}_{i,c,k} = C_{i,c} \cdot j \cdot 10^k \quad \forall c \end{bmatrix} \quad \forall i, k \in \{\psi, \ldots, \eta\} \tag{43}$$

Applying the convex hull reformulation leads to Eqs. (44) and (45).

$$\begin{aligned} \tilde{F}_{i,k} &= \sum_{j=0}^{9} j \cdot 10^k \cdot Y^F_{i,j,k} \quad \forall i, k \in \{\psi, \ldots, \eta\} \\ \tilde{Z}_{i,c,k} &= \sum_{j=0}^{9} \widehat{C}_{i,c,j,k} \cdot j \cdot 10^k \quad \forall i, c, k \in \{\psi, \ldots, \eta\} \end{aligned} \tag{44}$$

$$\begin{aligned} C_{i,c} &= \sum_{j=0}^{9} \widehat{C}_{i,c,j,k} \quad \forall i, c, k \in \{\psi, \ldots, \eta\} \\ \underline{c}_{i,c} \cdot Y^F_{i,j,k} &\le \widehat{C}_{i,c,j,k} \le \bar{c}_{i,c} \cdot Y^F_{i,j,k} \quad \forall i, c, j \in \{0, \ldots, 9\}, \\ &\quad k \in \{\psi, \ldots, \eta\} \\ \sum_{j=0}^{9} Y^F_{i,j,k} &= 1 \quad \forall i, k \in \{\psi, \ldots, \eta\} \end{aligned} \tag{45}$$

The lower bounding MILP model is then composed of Eqs. (16), (45)–(51).

$$\begin{aligned} F_i &= \sum_{k=\psi}^{\eta} \sum_{j=1}^{9} j \cdot 10^k \cdot Y^F_{i,j,k} + \ddot{F}_i \quad \forall i \\ Z_{i,c} &= \sum_{k=\psi}^{\eta} \sum_{j=1}^{9} \widehat{C}_{i,c,j,k} \cdot j \cdot 10^k + \ddot{Z}_{i,c} \quad \forall i, c \end{aligned} \tag{46}$$

$$\begin{aligned} &\underline{c}_{i,c} \cdot \ddot{F}_i \le \ddot{Z}_{i,c} \le \bar{c}_{i,c} \cdot \ddot{F}_i \quad \forall i, c \\ &(C_{i,c} - \bar{c}_{i,c}) \cdot 10^{\psi} + \bar{c}_{i,c} \cdot \ddot{F}_i \le \ddot{Z}_{i,c} \le (C_{i,c} - \underline{c}_{i,c}) \cdot 10^{\psi} \\ &\quad + \underline{c}_{i,c} \cdot \ddot{F}_i \quad \forall i, c \end{aligned} \tag{47}$$

$$\underline{f}_i \le F_i \le \bar{f}_i \quad \forall i \tag{48}$$

$$0 \le \ddot{F}_i \le 10^{\psi} \quad \forall i \tag{49}$$

$$\widehat{C}_{i,c,j,k} \ge 0 \quad \forall i, c, j \in \{0, \ldots, 9\}, \quad k \in \{\psi, \ldots, \eta\} \tag{50}$$

$$Y^F_{i,j,k} = \{0, 1\} \quad \forall i, j \in \{0, \ldots, 9\}, \quad k \in \{\psi, \ldots, \eta\} \tag{51}$$

6. Further details for relaxation of NLP design problem

We now highlight a couple of aspects involved in the generation of the alternative MILP relaxations for the reformulated NLP model discussed in Section 3. The first relevant aspect concerns the lower and upper bounds for the concentration and flowrate variables, generically represented by $\underline{f}_i$, $\bar{f}_i$, $\underline{c}_{i,c}$, and $\bar{c}_{i,c}$ that appear for instance in Eqs. (16), (47) and (48). These bounds are independent of the chosen discretization level and were calculated through Eqs. (52)–(56), which result from system analysis. While such bounds can be strengthened following minimization/maximization of each variable subject to the relaxed linear problem constraints (Ruiz & Grossmann, 2011), given in this case by Eqs. (2), (3), (7) and (9)–(16), this was not done in this work. In other words, no bound contraction methods are being used as a pre-processing step. It should also be mentioned that the exact same bounds are used

when solving the original NLP (Section 2.1) with local or global optimization solvers.

$$\underline{f}_{i,i'}^{U-U} = \underline{f}_{i}^{U-D} = 0 \quad \forall i \in I^{fl}, \quad i' \in I \tag{52}$$

$$\bar{f}_{i,i'}^{U-U} = \min\left(f_i^{U-\lim}, f_{i'}^{U-\lim}\Big|_{i' \in I^{fl}} + f_{i'}^{U-in}\Big|_{i' \in I^{ff}}\right) \quad \forall i \in I^{fl}, \quad i' \in I \tag{53}$$

$$\bar{f}_{i}^{U-D} = f_i^{U-\lim} \quad \forall i \in I^{fl} \tag{54}$$

$$\underline{c}_{i,c} = \underline{c}_{i,c}^{U-out} = \min_{w \in W}(c_{w,c}^{W}) + \Delta m_{i,c}^{U}/f_i^{U-\lim} \quad \forall i \in I^{fl}, \quad c \in C \tag{55}$$

$$\bar{c}_{i,c} = \bar{c}_{i,c}^{U-out} \quad \forall i \in I^{fl}, \quad c \in C \tag{56}$$

The second aspect involves the use of a tighter constraint for the relation between the disaggregated flowrate variables and the binary partition variables. Taking Eq. (28) as an example, the two sets of constraints become Eqs. (57) and (58). We can now sum constraint (57) over i' and add to Eq. (58) to get Eq. (59), which takes advantage of the fact that the upper bound of the outlet streams from splitter node SP_i is the same as the upper bound of its inlet flowrate, $f_i^{U-\lim}$. This is responsible for a significant increase in computational performance both for partitioning and parameterizing over the concentration variables.

$$\widehat{F}_{i,i',c,n}^{U-U} \le \bar{f}_{i,i'}^{U-U} \cdot Y_{i,c,n}^{C} \quad \forall i, i', c, n \tag{57}$$

$$\widehat{F}_{i,c,n}^{U-D} \le \bar{f}_{i}^{U-D} \cdot Y_{i,c,n}^{C} \quad \forall i, c, n \tag{58}$$

$$\sum_{i'} \widehat{F}_{i,i',c,n}^{U-U} + \widehat{F}_{i,c,n}^{U-D} \le f_i^{U-\lim} \cdot Y_{i,c,n}^{C} \quad \forall i, c, n \tag{59}$$

7. MDT relaxation for MINLP design problems

So far, we have focused on the design of water-using networks, which can be considered a subproblem of a fully integrated water network also featuring treatment units (Karuppiah & Grossmann, 2006). Since design problems from actual water networks should be formulated as mixed-integer nonlinear problems instead of NLPs, not necessary limited to nonlinear terms of the bilinear type, it is now worth to discuss how the relaxation from multiparametric disaggregation is affected.

7.1. Minimum flow constraints

To avoid overcomplicating the network, it may be convenient to choose a connection between units only if the flowrate exceeds a certain minimum value (f^{LO}). This will require defining additional binary variables (y) and constraints of type: $F \ge f^{LO} \cdot y$. Using these binary variables, one may also think of limiting the number of pipeline segments. Similar constraints result when selecting amongst alternative treatment technologies (see convex hull reformulation of the GDP model for the integrated problem in Karuppiah & Grossmann, 2006). These are linear constraints that can simply be added to those resulting from the MDT relaxation for the bilinear part of the problem, when transforming the original MINLP into a MILP.

7.2. Investment cost terms

The investment cost of treatment units is typically a function of type $\beta \cdot F^{\alpha}$. This term can be underestimated within a piecewise McCormick relaxation approach, as shown in Karuppiah and Grossmann (2006). In case α is a rational number (e.g. 0.6), multiparametric disaggregation can also provide a relaxation, after replacing the term by a new variable coupled with a constraint featuring monomials with natural exponents (see Teles et al., 2013). Note that bilinear terms form the basic building block to tackle higher order monomials.

7.3. Mixed-integer monomial terms

In general, multiparametric disaggregation can be applied to MINLPs of the mixed-integer polynomial type (Teles et al., 2012b). In fact, under the context of MDT, they can be considered a special case of polynomial problems. Whereas in a NLP, continuous variables being discretized will only assume the exact values for $\psi = -\infty$, for integer variables this will occur for $\psi = 0$. We can thus handle more realistic networks with other sources of nonlinear terms beyond the internal mixers, splitters and water-using units that were discussed in this paper.

8. Global optimization algorithm

Both the piecewise McCormick and the multiparametric disaggregation relaxation formulations are capable of generating a lower bound to the original NLP. This will become tighter and tighter as the number of partitions, in one case, and the discretization level, in the other case, increase at the expense of a linear/logarithmic increase in problem size (the proof that the lower bounding formulation from multiparametric disaggregation gives the global optimal solution in the limit of an infinite number of significant digits, is given in Kolodziej et al. (2012). Thus, with the exception of very small problems, it is not efficient to rely on the lower bounding formulations for a very high/low value of n/ψ to generate the global optimal solution. The sensible approach is to also use an upper bounding formulation and terminate whenever the optimality gap is below a given tolerance or upon reaching a specified maximum computational limit. As upper bounding approach, we consider the solution of the original NLP (Section 2.1) with a local solver, following initialization of the concentration and flowrate variables with the values from the lower bounding problem to ease convergence.

The proposed global optimization algorithm is given in Fig. 7. Given the relative optimality tolerance ε and the maximum computational limit CPU^{MAX} (and also the left truncation point η in the case of MDT), there are four alternatives for calculating the lower bound and initializing the flowrate and concentration variables. The first iteration of the piecewise McCormick formulations, featuring a single partition, is equivalent to using the global McCormick envelopes, whereas with the multiparametric disaggregation formulation $\psi = \eta$ may already lead to a few discrete points with respect to the approximation term of the parameterized variables, possibly resulting in a tighter lower bound. The maximum resource limit of the MILP solver is set through parameter *tilim*, where TCPU stands for time elapsed (an internal system variable). Provided that the original NLP is feasible, the relaxation model will not be integer infeasible and one gets a lower bound, LB, given by the best possible solution at the time of termination (OBJ.*est*). The values for the flowrate and concentration variables (attribute .*l*) are then used to initialize the NLP, the solution of which provides an upper bound (UB). With the upper bound, we can define a cutoff value for the next run of the MILP solver and calculate the optimality gap (GAP). Unless this is below the given tolerance ε or the maximum computational time has been reached, one proceeds to the next iteration.

For the next iteration, the accuracy of the partitioned/parameterized variables is increased by one significant digit, which corresponds to multiplying the number of partitions by 10 or having a single decrease in the value of ψ. The maximum resource limit is updated to ensure that the total computational time does not exceed CPU^{MAX} and the problem is resolved. Generally the lower bound will increase but, due to the cutoff value, it is possible that

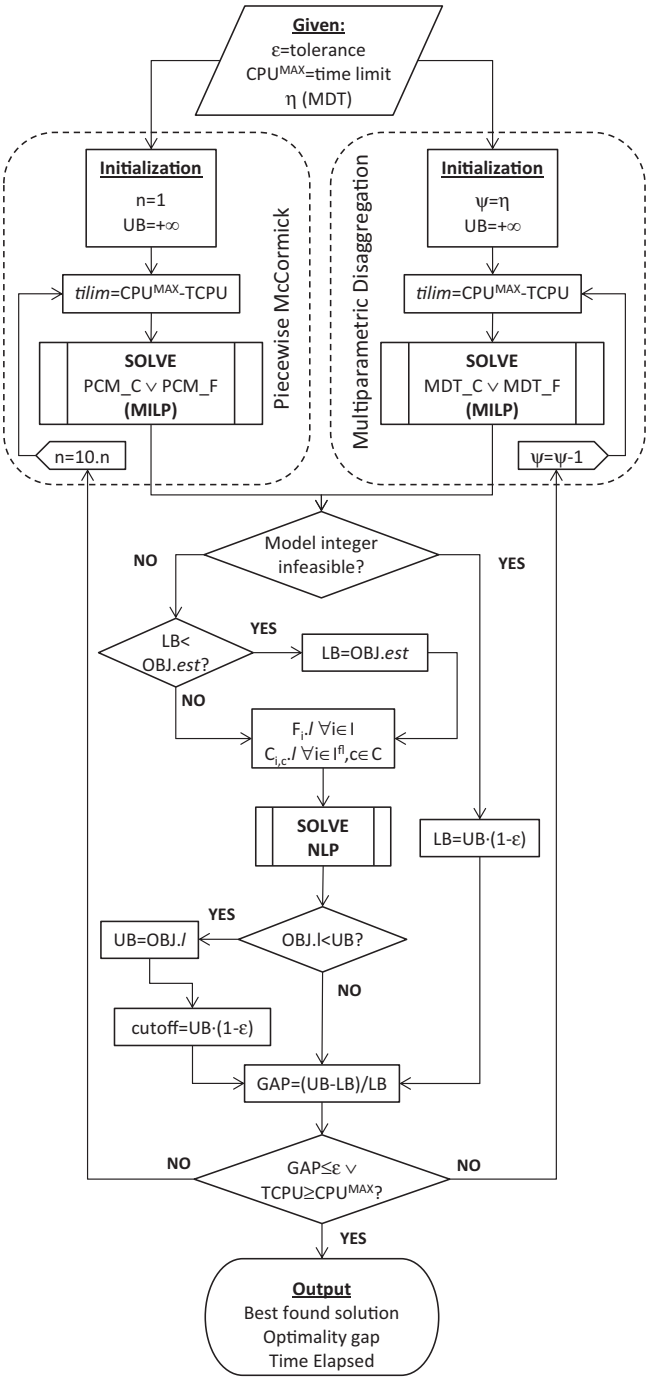

Fig. 7. Global optimization algorithm using either piecewise McCormick or multiparametric disaggregation relaxations.

all feasible solutions are removed from the feasible space. Once the solver realizes that no solution can be found, it terminates with an integer infeasible model status meaning that the current upper bound is within a relative tolerance ε of the global optimum. It is also possible for the solver to reach the maximum time limit without finding a better lower bound. Either way the search is over.

8.1. Remarks

The algorithm can start from any value of n or ψ since no information from previous iterations is required. This means that no variables related to the accuracy of the partitioned/discretized variables are being fixed for the next iteration.

For simplicity of presentation, we have assumed throughout the paper that the number of partitions/discretization level is the same for all partitioned/parameterized variables but it is straightforward to consider otherwise. This issue is particularly important for multiparametric disaggregation since even the same accuracy in terms of significant digits may correspond to different values of η and ψ (e.g. if different contaminants have concentrations of a different order of magnitude). This does not happen for the piecewise McCormick relaxation since the partitions normalize the variables domain.

9. Computational results

We consider 19 example problems (Ex1–Ex18 and Ex20) taken from Teles et al. (2009) to test the performance of the piecewise McCormick (PCM) and multiparametric disaggregation (MDT) approaches. Key performance indicators are given in Table 1, while the information related to the number and type of water-using units, freshwater sources and contaminants can be found in Table 2. Recall that attributes (_F) and (_C) identify the variables selected for partitioning/parameterization. The models have been implemented and solved in GAMS 23.8.2 using CPLEX 12.4 for the MILP and CONOPT 3 for the NLP problems, using a single thread and default options except for the 10^{-6} relative optimality tolerance. For the global optimization algorithm, the relative tolerance was set to $\varepsilon = 10^{-4}$ and the total computational time to $\text{CPU}^{\text{MAX}} = 3600$ CPUs. These same values were employed as termination criteria when solving the original NLP problems with the global optimization solvers BARON 10.2 (Sahinidis, 1996) and GloMIQO 1.0.0 (Misener & Floudas, 2012). The hardware consisted on an Intel Core i7 950@3.07 GHz processor with 8 GB of RAM running Windows 7 Professional.

9.1. Performance overview

Table 1 lists the optimality gap and total computational time for the four different formulations discussed earlier and for the multiparametric disaggregation concentrations approach involving a binary (MDT2_C) rather than decimal numeric representation system. These have been ordered by increasing number of successful runs (termination with relative tolerance criterion). In two of them (Ex1 and Ex4) global optimality was proven, meaning that the solution of the MILP for the initial value of ψ was exactly the same as the upper bound from CONOPT. The same result was obtained for the piecewise McCormick approach for $n = 1$. The results for the global optimization solvers GloMIQO and BARON are also part of Table 1. Notice that Ex20 is the only one that cannot be solved to 0.01% gap.

For interpretation of the results we use one of the standard metrics given in Dolan and Moré (2002) that can be used to generate performance profiles, a widely used tool for benchmarking and comparing optimization software. More specifically, being $t_{p,s}$ the performance measure for problem p, solver s (e.g. optimality gap, computational time), the performance ratio $r_{p,s}$ can be calculated by Eq. (60). To get the overall assessment of the performance of the solver we determine $\rho_s(\tau)$ as the probability that a performance ratio $r_{p,s}$ is within a factor $\tau \in \Re$ of the best possible ratio. The function ρ_s is the cumulative distribution function for the performance ratio. To allow for the analysis of the performance data over a wide range of τ values without overlooking the behavior for τ close to zero (probability that the solver is the best performer), we plot the results in a $\log_2$ scale, see Eq. (61), where n_p is the number of problems.

$$r_{p,s} = \frac{t_{p,s}}{\min_{s' \in S} t_{p,s'}} \quad \forall p \in P, \quad s \in S \tag{60}$$

$$\rho_s(\tau) = \frac{1}{n_p} \text{size}\{p \in P : \log_2(r_{p,s}) \leq \tau\} \quad \forall s \in S \tag{61}$$

Table 1
Optimality gap and total computational effort (runs meeting relative tolerance termination criterion in bold).

		Optimality gap						Total computational time (CPUs)						
	Optimum	PCM_F	PCM_C	MDT_F	MDT_C	MDT-2_C	GloMIQO	BARON	PCM_F	PCM_C	MDT_F	MDT_C	MDT-2_C	GloMIQO	BARON
Ex1	86.83333	**0.000%**	**0.000%**	**0.000%**	**0.000%**	**0.000%**	0.859%	0.629%	0.24	0.22	0.34	0.33	0.22	3600	3600
Ex2	74.46994	0.429%	0.216%	0.038%	**0.010%**	**0.010%**	0.167%	0.245%	3600	3600	3600	235	85.7	3600	3600
Ex3	143.4126	**0.010%**	**0.010%**	**0.010%**	**0.010%**	**0.010%**	0.303%	0.227%	105	184	9.27	9.67	7.75	3600	3600
Ex4	123.9286	**0.000%**	**0.000%**	**0.000%**	**0.000%**	**0.000%**	0.113%	0.333%	0.23	0.23	0.33	0.34	0.22	3600	3600
Ex5	197.6901	**0.010%**	**0.010%**	**0.010%**	**0.010%**	**0.010%**	**0.010%**	**0.010%**	124	0.60	9.39	0.61	0.69	0.43	1.09
Ex6	142.0816	0.222%	0.064%	**0.010%**	**0.010%**	**0.010%**	0.324%	0.250%	3600	3600	3600	124	614	3600	3600
Ex7	280.7712	0.087%	0.546%	0.019%	0.010%	**0.010%**	**0.010%**	**0.010%**	3600	3600	3600	3600	1818	12.0	453
Ex8	164.4898	1.847%	0.128%	0.025%	**0.010%**	**0.010%**	**0.010%**	**0.010%**	3600	3600	3600	44.7	29.6	8.12	18.9
Ex9	312.9215	0.636%	0.814%	0.133%	0.129%	0.086%	**0.010%**	0.211%	3600	3600	3600	3600	3600	26.9	3600
Ex10	169.1173	0.073%	0.057%	**0.010%**	**0.010%**	**0.010%**	**0.010%**	**0.010%**	3600	3600	72.5	11.5	3.21	5.10	373
Ex11	104.8861	0.794%	0.032%	**0.010%**	**0.010%**	**0.010%**	**0.010%**	**0.010%**	3600	3600	1889	77.7	30.2	390	2553
Ex12	165.1953	1.274%	0.161%	0.141%	**0.010%**	**0.010%**	0.543%	0.357%	3600	3600	3600	23.3	13.7	3600	3600
Ex13	178.2629	1.083%	0.573%	0.455%	0.160%	0.019%	**0.010%**	0.491%	3600	3600	3600	3600	3600	232	3600
Ex14	329.5698	2.359%	1.386%	3.254%	0.395%	0.215%	**0.010%**	2.009%	3600	3600	3600	3600	3600	1980	3600
Ex15	361.5177	2.057%	1.204%	2.774%	0.477%	0.276%	**0.010%**	1.523%	3600	3600	3600	3600	3600	1916	3600
Ex16	285.9343	1.196%	1.702%	0.023%	**0.010%**	**0.010%**	**0.010%**	**0.010%**	3600	3600	3600	1024	226	6.26	32.2
Ex17	157.0944	0.068%	**0.010%**	0.017%	**0.010%**	**0.010%**	3.139%	2.436%	3600	2.65	3600	3.61	2.93	3600	3600
Ex18	238.7333	0.060%	**0.010%**	**0.010%**	**0.010%**	**0.010%**	0.825%	0.984%	3600	32.2	517	2.11	2.43	3600	3600
Ex20	403.1960	2.975%	1.266%	3.978%	0.386%	0.185%	2.479%	3.750%	3600	3600	3600	3600	3600	3600	3600

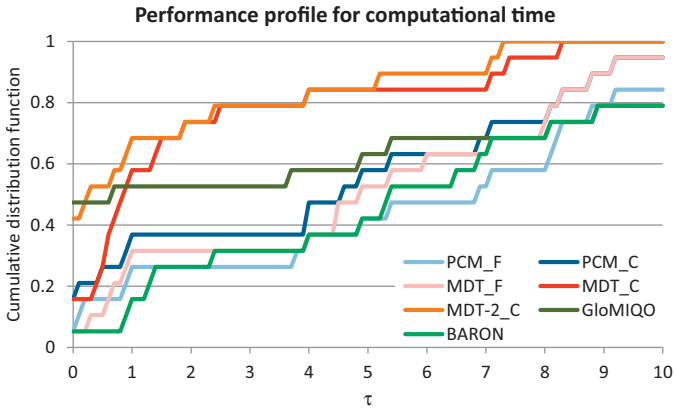

Fig. 8. Performance profile for computational time.

The performance profile for total computational time is given in Fig. 8. The results for $\tau = 0$ indicate that GloMIQO has the most wins with a probability of 47% that is the fastest solver on a given problem, soon followed by MDT2_C at 42%. On the other end, BARON, PCM_F and MDT_F are always beaten by some other solver/algorithm, with the 5% probability resulting from the single problem (Ex20) that cannot be solved within the given tolerance (all the solvers took the same time, 3600 CPUs). The cumulative distribution functions of MDT_C and MDT2_C surpass the one for GloMIQO for higher values of τ. The probability that MDT-2_C is at most twice as slow the fastest solver ($\tau = 1$) is already 84% while the values for GloMIQO and BARON are 53 and 32%, respectively. This tells us that the new multiparametric disaggregation approach for the alternative of parameterizing the concentrations is more competitive.

The performance is even better from the perspective of the optimality gap. After normalizing the absolute zero values to 0.010% to avoid problems in Eq. (60), Fig. 9 results. Now, MDT2_C has the most wins, 15 out of 19, some of them tied with other solvers. The probability that the optimality gap is at most one order of magnitude larger than the best solver ($\tau = 3.33$) is equal to 89% while for the commercial solvers we get 53% for GloMIQO and 31% for BARON. It can thus be concluded that: multiparametric disaggregation should be preferred over the piecewise relaxation; it is best to parameterize/partition the concentration variables; the binary system is slightly better than its decimal counterpart. When compared to the commercial solvers, and BARON in particular, it is fair to say that part of the success is due to the reformulation linearization technique that was applied to reduce the number of bilinear terms (as described in Section 3), which successfully made the relaxation

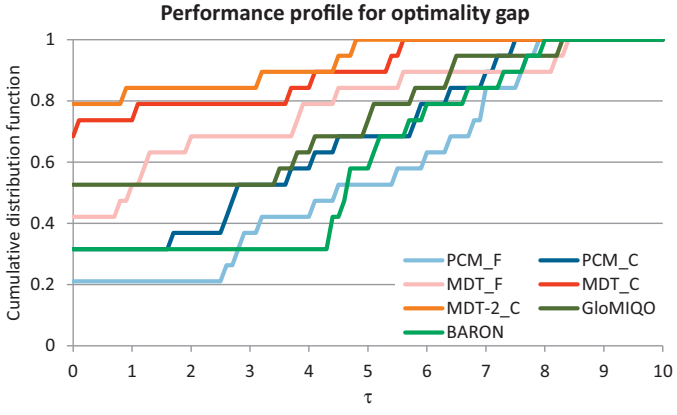

Fig. 9. Performance profile for optimality gap.

Table 2
Optimality gap in the first iteration of algorithm calculated using the global optimal solution as the upper bound and other statistics.

	Problem features				Optimality gap in first iteration					NLP problem size	
	$\|I^{fl}\|$	$\|I^{ff}\|$	$\|W\|$	$\|C\|$	PCM_C[a] (%)	MDT_F (%)	MDT_C (%)	GloMIQO (%)	BARON (%)	Variables	Equations
Ex1	3	1	1	2	0.0	0.0	0.0	35.9	768	40	23
Ex2	4	0	3	4	3.5	2.6	3.0	6.7	399	69	41
Ex3	4	0	2	4	0.7	0.7	0.7	2.2	245	65	41
Ex4	4	2	1	2	0.0	0.0	0.0	28.7	485	69	33
Ex5	4	3	2	2	0.4	0.4	0.4	0.0	97.7	91	37
Ex6	5	0	4	5	1.5	1.2	1.5	3.6	35.0	106	61
Ex7	5	0	3	6	4.7	4.7	4.7	2.3	131	111	71
Ex8	5	0	3	4	12.6	12.6	12.6	21.5	128	91	51
Ex9	6	0	2	6	4.1	4.1	4.1	2.7	219	133	85
Ex10	3	1	2	3	3.4	3.4	3.4	3.6	170	50	30
Ex11	3	1	4	4	8.3	4.5	8.3	21.7	133	64	37
Ex12	4	0	3	2	15.0	15.0	11.4	24.5	22.6	53	25
Ex13	4	2	3	4	6.1	1.8	6.0	4.9	540	97	53
Ex14	4	3	2	6	6.8	6.8	6.6	6.0	264	123	81
Ex15	5	2	2	6	6.3	6.2	6.1	5.6	361	136	87
Ex16	4	1	3	4	8.8	8.8	4.8	10.1	178	82	47
Ex17	4	1	3	3	3.8	1.1	3.8	11.3	502	74	38
Ex18	3	1	3	4	1.8	1.8	1.8	10.8	166	60	37
Ex20	4	5	4	5	6.8	6.8	3.3	7.3	55.4	171	84

[a] PCM_F leads to the same results except for Ex9 (4.9%) and Ex14 (6.9%).

tighter, see Table 2. As an example, the optimality gap in the first iteration of Ex20 for MDT_C is lower than the one from BARON at termination.

9.2. Multiparametric disaggregation vs. piecewise relaxation

The results in Table 2 for the piecewise relaxation approach represent the linear relaxation gap, whereas a few binary variables may already be involved in the first iteration of the multiparametric disaggregation approaches. This is reason why the latter is sometimes tighter at the expense of a higher computational time (first iteration of MDT_C for Ex20 ends at 5.41, compared to 0.37 CPUs of PCM_C, see Table 3). Notice that there are certain differences between the two parameterization schemes and that despite being tighter in five cases, MDT_F performed worse than MDT_C in all of them. It should also be highlighted that the base used for numeric representation does not affect the quality of the relaxation.

Table 3
Key computational statistics for different iterations.

		Ex2	Ex14	Ex15	Ex18	Ex20
Optimality gap						
MDT_C	$\psi = 2$	3.028%	6.567%	6.077%	1.832%	3.307%
	$\psi = 1$	1.106%	2.042%	1.755%	0.399%	0.386%
	$\psi = 0$	0.134%	0.395%	0.477%	0.010%	0.386%
MDT_C no Eq. (16)	$\psi = 2$	5.091%	19.04%	18.03%	1.832%	12.80%
	$\psi = 1$	4.968%	3.284%	3.031%	0.399%	0.593%
	$\psi = 0$	0.134%	2.386%	1.051%	0.010%	0.593%
PCM_C	$n = 1$	3.496%	6.768%	6.260%	1.832%	6.766%
	$n = 10$	0.216%	1.386%	1.204%	0.075%	1.266%
	$n = 100$	0.216%	1.386%	1.204%	0.010%	1.266%
Computational time (CPUs)						
MDT_C	$\psi = 2$	0.34	0.61	0.69	0.23	5.41
	$\psi = 1$	0.91	35.0	43.3	0.69	1030
	$\psi = 0$	10.6	3600	3600	2.11	3600
MDT_C no Eq. (16)	$\psi = 2$	0.46	0.94	1.33	0.27	2.52
	$\psi = 1$	1.36	171	738	0.84	1461
	$\psi = 0$	57.4	3600	3600	2.87	3600
PCM_C	$n = 1$	0.22	0.52	0.31	0.22	0.37
	$n = 10$	4.61	63.6	50.1	0.70	71.8
	$n = 100$	3600	3600	3600	32.2	3600
Binary variables						
MDT_C	$\psi = 2$	18	34	45	14	65
	$\psi = 1$	70	205	276	58	251
	$\psi = 0$	230	445	576	178	451
	$\psi = -1$	390	685	876	298	651
PCM_C	$n = 1$	16	24	30	12	20
	$n = 10$	160	240	300	120	200
	$n = 100$	1600	2400	3000	1200	2000
Variables						
MDT_C	$\psi = 0$	1641	4560	5860	1272	5572
PCM_C	$n = 100$	11,365	24,339	30,406	8532	24,391
Equations						
MDT_C	$\psi = 0$	1163	2683	3384	887	2833
MDT_C no Eq. (16)	$\psi = 0$	843	1915	2424	647	2033
PCM_C	$n = 100$	5305	8337	10,407	3985	7164

To help explain the difference in performance between the algorithms based on multiparametric disaggregation and piecewise relaxations, we show in Table 3 the progress of optimality gap, cumulative computational time and binary variables through three iterations for five representative examples. For multiparametric disaggregation, we also test the impact of removing the global McCormick envelopes, Eq. (16), from the formulation. In this respect, the results clear show that these are important constraints, with the benefits from a tighter linear relaxation (e.g. 3.307 vs. 12.80% for Ex20 and $\psi = 2$) on the computational time and optimality gap at termination, far exceeding the difficulties from solving problems with a considerably larger number of constraints (roughly 40% more for $\psi = 0$).

The first two iterations can be solved rather fast, roughly in less than a minute, with PCM_C actually leading to lower optimality gaps indicating that it is a tighter approach than MDT_C ($\psi = 2$ should be compared to $n = 1$, $\psi = 1$ with $n = 10$ and $\psi = 0$ with $n = 100$). However, increasing the number of partitions by one order of magnitude makes the problem too difficult to solve except for Ex18, with no benefit in optimality gap, meaning that the best possible solution from the MILP at termination is worse than the lower bound from the previous iteration. In contrast, MDT_C can still improve substantially the lower bound in four cases, including two for which it cannot prove optimality (Ex14 and Ex15). And the reason can be seen in the binary variables rows. While multiplying the number of partitions by 10 leads to the same increase in binary variables for PCM_C, for MDT_C we get a logarithmic rather than linear increase in problem size with the number of partitions, and thus the MILP problem remains tractable longer. Similar remarks can be made concerning the number of total variables and constraints. Finally, it should be highlighted that the best results for multiparametric disaggregation involve MILP problem sizes that are typically between one to two orders of magnitude those of the original NLP (compare the last rows of Table 3 with the two last columns of Table 2).

10. Conclusions

This paper has proposed global optimization algorithms for the design of water-using networks, a nonlinear problem of the bilinear type involving the product of flowrates by concentrations. The main novelty has been the derivation of mixed-integer linear programming relaxations using our recently developed multiparametric disaggregation concept starting from the elementary disjunctive programming models. When compared to their piecewise McCormick counterparts, they exhibited a considerably better computational performance due to a more favorable scaling of problem size with increasing accuracy. The results on a set of nineteen example problems from the literature have shown that the best choice of parameterized variables lies with the concentrations and that its more efficient to use the binary instead of the decimal numeric representation system. Through standard performance profile plots of the computational time and optimality gap indicators, it has been shown that such algorithm outperforms commercial solvers BARON and GloMIQO for this specific class of chemical engineering problems.

While multiparametric disaggregation has been shown quite competitive as a standalone approach, given the large increase in problem size with the number of parameterized variables and accuracy level (with respect to the original nonlinear problem), it may quickly loose efficiency, particularly when moving from bilinear to polynomial problems. Future work will thus look at the integration with spatial branch and bound methods, applying the new method after branching on the variables domain hoping that the resulting smaller and tighter subproblems can be solved faster than the full problem.

Acknowledgments

The authors gratefully acknowledge financial support from the Luso-American and National Science Foundations, under the 2011 Portugal-U.S. Research Networks Program.

References

Bagajewicz, M. (2000). A review of recent design procedures for water networks in refineries and process plants. *Computers & Chemical Engineering*, *24*, 2093–2113.
Balas, E. (1979). Disjunctive programming. *Annals of Discrete Mathematics*, *5*, 3–51.
Balas, E. (1985). Disjunctive programming and a hierarchy of relaxations for discrete optimization problems. *SIAM Journal of Algebraic and Discrete Mathematics*, *6*, 466–486.
Bergamini, M. L., Aguirre, P., & Grossmann, I. E. (2005). Logic-based outer approximation for globally optimal synthesis of process networks. *Computers and Chemical Engineering*, *29*, 1914–1933.
Castro, P. M., & Grossmann, I. E. (2012). Generalized disjunctive programming as a systematic modeling framework to derive scheduling formulations. *Industrial and Engineering Chemistry Research*, *51*, 5781–5792.
Dolan, E. D., & Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming Series A*, *91*, 201–213.
El-Halwagi, M. M., Glaswog, I. M., Qin, X., & Eden, M. R. (2004). Property integration: Componentless design techniques and visualization tools. *AIChE Journal*, *50*(8), 1854–1869.
Faria, D. C., & Bagajewicz, M. J. (2012a). A new approach for global optimization of a class of MINLP problems with applications to water management and pooling problems. *AIChE Journal*, *58*(8), 2320–2335.
Faria, D. C., & Bagajewicz, M. J. (2012b). Global optimization based on subspaces elimination: Applications to generalized pooling and water management problems. *AIChE Journal*, *58*(8), 2336–2345.
Gounaris, C. E., Misener, R., & Floudas, C. A. (2009). Computational comparison of piecewise-linear relaxations for pooling problems. *Industrial and Engineering Chemistry Research*, *48*, 5742–5766.
Hasan, M. M. F., & Karimi, I. A. (2010). Piecewise linear relaxation of bilinear programs using bivariate partitioning. *AIChE Journal*, *56*, 1880.
Haverly, C. A. (1978). Studies of the behavior of recursion for the pooling problem. *SIGMAP Bulletin*, *25*, 19–28.
Jezowski, J. (2010). Review of water network design methods with literature annotations. *Industrial & Engineering Chemistry Research*, *49*, 4475–4516.
Karuppiah, R., & Grossmann, I. E. (2006). Global optimization for the synthesis of integrated water systems in chemical processes. *Computers and Chemical Engineering*, *30*, 650–673.
Kolodziej, S., Castro, P. M., & Grossmann, I. E. (2012). Global optimization of bilinear programs with a multiparametric disaggregation technique. *Journal of Global Optimization*, http://dx.doi.org/10.1007/s10898-012-0022-1
Liberti, L., & Pantelides, C. C. (2006). An exact reformulation algorithm for large nonconvex NLPs involving bilinear terms. *Journal of Global Optimization*, *36*, 161–189.
McCormick, G. P. (1976). Computability of global solutions to factorable nonconvex programs. Part I. Convex underestimating problems. *Mathematical Programming*, *10*, 147–175.
Meyer, C. A., & Floudas, C. A. (2006). Global optimization of a combinatorially complex generalized pooling problem. *AIChE Journal*, *52*(3), 1027–1037.
Misener, R., & Floudas, C. A. (2010). Global optimization of large-scale generalized pooling problems: Quadratically constrained MINLP models. *Industrial and Engineering Chemistry Research*, *49*, 5424.
Misener, R., & Floudas, C. A. (2012). GloMIQO: Global mixed-integer quadratic optimizer. *Journal of Global Optimization*, http://dx.doi.org/10.1007/s10898-012-9874-7
Misener, R., Thompson, J. P., & Floudas, C. A. (2011). APOGEE: Global optimization of standard, generalized, and extended pooling problems via linear and logarithmic partitioning schemes. *Computers & Chemical Engineering*, *35*, 876–892.
Nápoles-Rivera, F., Ponce-Ortega, J. M., El-Halwagi, M. M., & Jiménez-Gutiérrez, A. (2010). Global optimization of mass and property integration networks with in-plant property interceptors. *Chemical Engineering Science*, *65*, 4363–4377.
Prakash, R., & Shenoy, U. (2005). Targeting and design of water networks for fixed flowrate and fixed contaminant load operations. *Chemical Engineering Science*, *60*, 255–268.
Quesada, I., & Grossmann, I. E. (1995). Global optimization of bilinear process networks with multicomponent flows. *Computers & Chemical Engineering*, *19*, 1219–1242.
Rubio-Castro, E., Ponce-Ortega, J. M., Serna-González, M., El-Halwagi, M. M., & Pham, V. (2012). Global optimization in property-based inter-plant water integration. *AIChE Journal*, http://dx.doi.org/10.1002/aic.13874
Ruiz, J. P., & Grossmann, I. E. (2011). Exploiting vector space properties to strengthen the relaxation of bilinear programs arising in the global optimization of process networks. *Optimization Letters*, *5*, 1.
Ryoo, H. S., & Sahinidis, N. V. (1996). A branch-and-reduce approach to global optimization. *Journal of Global Optimization*, *8*, 107–138.
Sahinidis, N. (1996). BARON: A general purpose global optimization software package. *Journal of Global Optimization*, *8*, 201–205.
Sherali, H., & Alameddine, A. (1992). A new reformulation–linearization technique for bilinear programming problems. *Journal of Global Optimization*, *2*, 379–410.

Teles, J. P., Castro, P. M., & Matos, H. A. (2013). Multiparametric disaggregation technique for global optimization of polynomial programming problems. *Journal of Global Optimization*, *55*, 227–251.

Teles, J. P., Castro, P. M., & Matos, H. A. (2012a). Global optimization of water networks design using multiparametric disaggregation. *Computers and Chemical Engineering*, *40*, 132–147.

Teles, J. P., Castro, P. M., & Matos, H. A. (2012b). Univariate parameterization for global optimization of mixed-integer polynomial problems. *European Journal of Operational Research*,. Under review.

Teles, J. P., Castro, P. M., & Novais, A. Q. (2008). LP-based solution strategies for the optimal design of industrial water networks with multiple contaminants. *Chemical Engineering Science*, *63*, 376–394.

Teles, J. P., Castro, P. M., & Novais, A. Q. (2009). MILP-based initialization strategies for the optimal design of water-using networks. *Chemical Engineering Science*, *64*, 3736–3752.

Vielma, J. P., & Nemhauser, G. (2011). Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. *Mathematical Programming*, *128*, 49–72.

Vielma, J. P., Ahmed, S., & Nemhauser, G. (2010). Mixed-integer models for nonseparable piecewise-linear optimization: Unifying framework and extensions. *Operations Research*, *58*, 303–315.

Wicaksono, D. N., & Karimi, I. A. (2008). Piecewise MILP under- and overestimators for global optimization of bilinear programs. *AIChE Journal*, *54*, 991–1008.