Global Optimization in Property-Based Interplant Water Integration

Eusiel Rubio-Castro
Dept. of Chemical Engineering, Universidad Michoacana de San Nicolás de Hidalgo, Morelia, Michoacán 58060, México

Dept. of Chemical and Biological Sciences, Universidad Autónoma de Sinaloa, Culiacán, Sinaloa 80000, México

José María Ponce-Ortega and Medardo Serna-González
Dept. of Chemical Engineering, Universidad Michoacana de San Nicolás de Hidalgo, Morelia, Michoacán 58060, México

Mahmoud M. El-Halwagi
Dept. of Chemical Engineering, Texas A&M University, College Station, TX 77843

Dept. of Chemical and Materials Engineering, King Abdulaziz University, Jeddah, Saudi Arabia

Viet Pham
Dept. of Chemical Engineering, Texas A&M University, College Station, TX 77843

DOI 10.1002/aic.13874
Published online July 3, 2012 in Wiley Online Library (wileyonlinelibrary.com).

This article presents a new global optimization method for the interplant water integration based on properties to characterize streams with numerous components. The problem is formulated as an mixed-integer non-linear programming (MINLP) model based on a superstructure that involves all possible options of interest (i.e., reuse and recycle in the same and to other plants and a set of shared treatment units). This formulation exhibits multiple local minima, and to overcome this problem, this article proposes effective branching rules in addition to two new reformulations for the upper bound (integer feasible solution) and the lower limit (relaxed solution), which are incorporated into a spatial branch and bound procedure to handle the bilinear terms in the model. The objective consists in finding the configuration with the minimum total annual cost. Results show that the global optimal solution (involving significant reductions in the fresh water consumption) is reached in few iterations and short central processing unit (CPU) time. © 2012 American Institute of Chemical Engineers *AIChE J,* 59: 813–833, 2013

Keywords: interplant water integration, deterministic global optimization, property integration, eco-industrial parks, bilinear terms

Introduction

Nowadays, the current situation in the industry around the world has demanded the development of efficient strategies for reducing simultaneously the environmental impact and the associated costs to yield sustainable industrial processes. In this regard, water is probably the most intensively used resource in the process industries and, in recent years, several works have addressed the problem for the efficient use of water considering the recycle, reuse, and regeneration for this important resource in single industrial facilities (i.e., the single plant water integration). The review articles by Bagajewicz,[1] Dunn and El-Halwagi,[2] and Foo[3] summarize the methodologies reported for the single plant water integration. Because significant environmental and economic benefits have been observed by implementing water integration in individual plants, this approach has been extended to study the water integration between multiple plants in eco-industrial parks (EIP) or interplant water integration. In this regard, the interplant water integration represents additional savings in the fresh water consumption and additional reductions in the overall wastewater discharged to the environment, in addition to the economic benefits for the use of a shared treatment system respect to the single plant integration.

The reported methodologies for interplant water integration can be classified as graphical, algebraic, and mathematical programming-based approaches. The articles by Olesen and Polley[4] and Spriggs et al.[5] address the interplant water integration using graphical methods based on the water pinch technology. On the other hand, the articles by Chew et al.,[6] Foo,[7] Bandyopadhyay et al.,[8] Chew et al.[9,10] present algebraic approaches for targeting minimum fresh water and waste flow rates for interplant water networks. The major advantage of the graphical and algebraic methods is that they provide targets for water integration before the

Correspondence concerning this article should be addressed to J. M. Ponce-Ortega at jmponce@umich.mx.

synthesis stage; however, these methodologies are restricted to single contaminants and cannot handle topological constraints. Furthermore, they are based on a sequential approach that does not guarantee the global optimal solution in the design stage.

Mathematical programming techniques can be used to solve large problems (i.e., with a large number of plants, process sources, and process sinks), including several pollutants as well as topological and operational constraints. In this context, several articles have addressed the synthesis problem of interplant water networks based on the composition of the streams (see, e.g., Lovelady et al.;[11] Liao et al.;[12] Chew et al.;[13] Chew and Foo;[14] Lovelady and El-Halwagi;[15] Lim and Park;[16] Chen et al.;[17] Aviso et al.;[18,19] Rubio-Castro et al.;[20,21] Tan et al.;[22] Taskhiri et al.[23]). Recently, Lovelady et al.[24] addressed the optimal design of interplant water networks where multiple pollutants are considered. In that work, the water streams are characterized in terms of the properties that affect the process sinks and that are restricted by the environmental regulations considering the property balances previously used by Ng et al.,[25] Ponce-Ortega et al.[26,27] and Nápoles-Rivera et al.[28] However, in general, all the aforementioned works for the interplant water integration have the following three major limitations:

- **Simplified superstructures**. To avoid numerical complications, several potentially good possibilities such as the optimal selection of the type of treatment units and different options for water integration (i.e., direct interplant integration, direct discharge to the environment, etc.) have not been considered by the earlier approaches. In addition, the environmental constraints have not been taken into account properly to avoid further numerical problems in the optimization procedures. In fact, it is worth pointing out that in the previous published articles all possible options and aspects of an interplant water integration are not considered such as direct flow rates between plants, environmental constraints, fixed cost for the interceptors, optimal selection of interceptors (Chew et al.,[13] Lovelady et al.[24]), multiple pollutants (Liao et al.[12]), cross-plant pipeline cost, and regeneration cost (Lim and Park[16]) as well as recycling streams between interceptors (Rubio-Castro et al.[20,21]). As a result, the simplified superstructures and their associated formulations may yield suboptimal solutions.
- **Composition-based integration**. It is very difficult to characterize wastewater streams that are constituted by numerous pollutants using only the composition. Additionally, the process and environmental constraints usually are stated in terms of specific properties (i.e., pH, density, viscosity, toxicity, color, chemical oxygen demand, etc.). Therefore, the composition-based interplant water integration may be not useful for systems with multiple contaminants that require the consideration of properties in addition to concentrations. For these cases, the property integration framework that was developed for individual water networks (Shelley and El-Halwagi[29]) can be extended for the interplant water integration problem. In this regard, Lovelady et al.[24] extended the scheme reported by Chew et al.[13] to include property-based constraints. Also, Chew and Foo[14] presented the automated targeting technique for interplant water integration that is based on the concept of pinch analysis. This technique is formulated as a linear programming model and enables the setting of various network targets before detailed design. Chew et al.[9] later proposed a new algorithm for targeting minimum fresh resource and waste flow rates for an interplant resource conservation network. These targeting algorithms are applicable for property-based interplant water networks. This article expands the scope of the previous research efforts in this area by introducing a general superstructure and a global optimization approach for the design of property-based interplant water networks.
- **Do not guarantee the global optimal solution**. Most of the previously reported methods for the interplant water integration have not guaranteed to find the global optimum and have used local optimization procedures (Lim and Park,[16] Chen et al.,[17] Liao et al.;[12] Aviso et al.;[18] Tan et al.[22]) such as the solver DICOPT (Viswanathan and Grossmann[30]). However, the model formulations for the interplant water integration are highly nonconvex because the large number of bilinear terms in the balance equations and the concave cost terms in objective functions. Therefore, when these previously reported approaches are used, it is very difficult to get the global optimal solution and even to get a feasible solution. In contrast to the previous works, Chew et al.[13] and Rubio-Castro et al.[20,21] addressed the optimal synthesis of an interplant water network by using global searches based on the linearization technique by Quesada and Grossmann[31] and the discretization approach by Pham et al.,[32] respectively. However, as shown above, the source-interception-sink structural representation that was proposed by these authors has not embedded all potential configurations and options of interest, and these were based on the stream compositions.

Notice that none of the previously reported works for the interplant water integration simultaneously meets the three issues indicated above. Therefore, the major contribution of this article is to overcome all these limitations using a general superstructure and a global optimization approach for the design of property-based interplant water networks. The superstructure considers the wastewater reuse in the same plant, the water exchange with others plants, and recycling streams between a shared set of available interceptors whose selection is simultaneously optimized to satisfy the environmental and process constraints based on the properties of the streams. It is noteworthy that wastewater reuse is made from the scheme presented in Appendix A. The new model based on the proposed superstructure includes the capital and operational costs for the interceptors, the cross-plant pipeline cost, and the fresh water cost, which form the total annual cost to minimize. The mass water integration is based on several properties such as composition, toxicity, pH, chemical oxygen demand, and others. Also, the model is extended to include property interceptors and the constraints for the process units and the environment are given in terms of such properties. The model includes property mixing rules based on the property operators shown in Table 1 (El-Halwagi et al.;[33] Shelley and El-Halwagi[29]). To obtain the global optimal solution for this problem, this article proposes a new strategy based on a spatial branch and bound scheme using new formulations for the lower and upper bounds for the iterative process in addition to provide branching rules for this specific problem, which helps to reduce significantly the number of iterations and the central processing unit (CPU) time for this type of problems that involve a large set of bilinear terms. In addition, the nonconvex terms in the objective function have been convexified. The following assumptions are considered to solve this problem: first, as shown in Table 1, ideal property operators are used to describe mixing of streams with different properties, in which the final

Table 1. Some Mixing Property Operator

Property	Operator
Composition	$\psi_z(z) = z$
Toxicity	$\psi_{tox}(tox) = tox$
Chemical oxygen demand	$\psi_{COD}(COD) = COD$
pH	$\psi_{pH}(pH) = 10^{pH}$
Density	$\psi_\rho(\rho) = 1/\rho$
Viscosity	$\psi_\mu(\mu) = \log(\mu)$
Reid vapor pressure	$\psi_{RVP}(RVP) = RVP^{1.44}$
Electric resistivity	$\psi_R(R) = 1/R$
Paper reflectivity	$\psi_{R\infty}(R_\infty) = R_\infty^{5.92}$
Color	$\psi_{color}(color) = color$
Odor	$\psi_{odor}(odor) = odor$

property is proportional to the mass contributions of the streams being mixed; in addition, constant efficiency factors are used to model the treatment units for the different properties.

The article is organized as follows. First the model formulation is presented. Then, the proposed global optimization approach to solve the model is outlined, and two case studies are presented to illustrate the performance of the proposed synthesis method. Finally, the conclusions are discussed, and the used nomenclature is presented, as well as the Appendixes A and B.

Model Formulation

The proposed model is based on the superstructure shown in Figure 1, which considers the reuse and recycle of the sources to the process sinks of the same plant as well as to the process sinks of different plants, in addition to the shared treatment units to treat the properties of the streams to meet the constraints on the process sinks and the environmental regulation for the discharge. Notice that each process source can be sent to each process sink in the same and/or other plant, to each interceptor and to the waste that is discharged to the environment. In addition, the flow rate from each interceptor can be segregated and sent to each process sink, to other interceptors, and to the waste discharged to the environment. One fictitious interceptor is used when no treatment is required, and also it takes into account the direct recycle for the process sources. Appendix A shows the explanation for the different reuse schemes presented in the proposed superstructure.

The goal of the proposed model consists in finding the configuration of the interplant water network that represents the minimum total annual cost and meets the operational and environmental constraints.

In the next equations, the subscripts i, j, k, r, w, t, p, q^1, q^2 and z are used to denote the process sources, process sinks, any flow rate, property interceptors, type of fresh

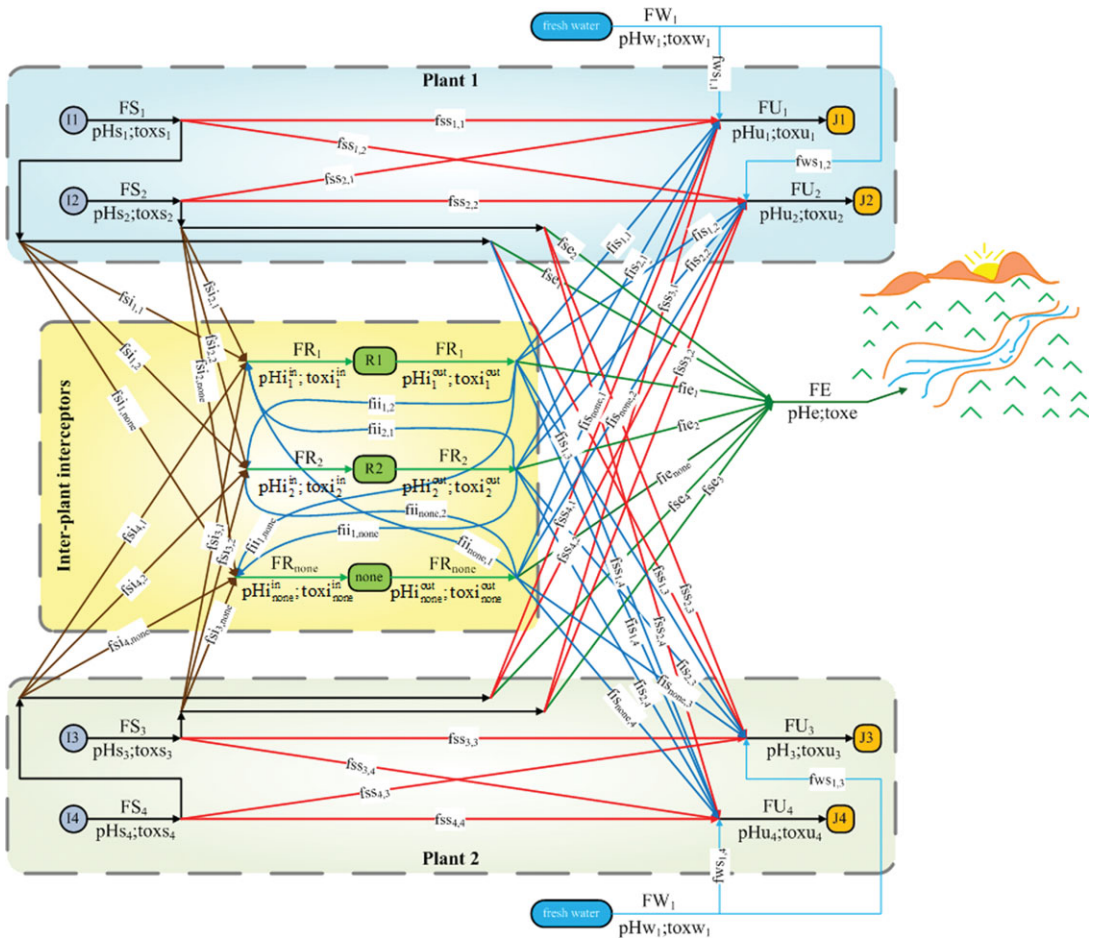

Figure 1. Superstructure proposed.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com]

water, iterations, properties, partitions for the upper bound, partitions for the lower bound, and linear segments to linearize the objective function, respectively; the superscript min represents a lower bound and the superscript max represents an upper bound. In addition, the variables in the model are italic letters and the parameters are regular letters. The equations of the proposed model are presented as follows.

Objective function

The objective function consists in minimizing the total annual cost (*TAC*) that includes the fresh water cost (*WC*), the regeneration cost (*RC*), and the cross-plant pipeline capital cost (*PC*):

$$TAC = WC + RC + PC \tag{1}$$

The fresh water cost is calculated with the next relationship,

$$WC = \mathrm{H_Y} \sum_{\mathrm{j}=1}^{\mathrm{J}} \sum_{\mathrm{w}=1}^{\mathrm{W}} fws_{\mathrm{w,j}} \mathrm{CU_w} \tag{2}$$

On the other hand, the cross-plant pipeline capital cost is a function of pipeline length, flow rate on pipeline, fluid density, velocity, type of pipe material, and so forth. For example, the Eq. 3 was adapted for interplant integration by Chew et al.[13] from Kim and Smith.[34] Notice in this equation that the capital cost of the pipelines is correlated with cross-sectional area of piping and it includes the fixed and variable costs for all pipe segments.

$$PC = \mathrm{K_F} \left\{ \begin{array}{l} \mathrm{p} \sum_{\mathrm{i}=1}^{\mathrm{I}} \sum_{\mathrm{j}=1}^{\mathrm{J}} \frac{\mathrm{D}_{\mathrm{i,j}}^{1} fss_{\mathrm{i,j}}}{3600\rho \mathrm{v}} + x_{\mathrm{i,j}}^{1} \mathrm{D}_{\mathrm{i,j}}^{1} \mathrm{CU_p} + \mathrm{p} \sum_{\mathrm{i}=1}^{\mathrm{I}} \sum_{\mathrm{r}=1}^{\mathrm{R}} \frac{\mathrm{D}_{\mathrm{i,r}}^{2} fsi_{\mathrm{i,r}}}{3600\rho \mathrm{v}} + x_{\mathrm{i,r}}^{2} \mathrm{D}_{\mathrm{i,r}}^{2} \mathrm{CU_p} + \\ \mathrm{p} \sum_{\mathrm{r}=1}^{\mathrm{R}} \sum_{\mathrm{j}=1}^{\mathrm{J}} \frac{\mathrm{D}_{\mathrm{r,j}}^{3} fis_{\mathrm{r,j}}}{3600\rho \mathrm{v}} + x_{\mathrm{r,j}}^{3} \mathrm{D}_{\mathrm{r,j}}^{3} \mathrm{CU_p} + \mathrm{p} \sum_{\mathrm{r}=1}^{\mathrm{R}} \frac{\mathrm{D}_{\mathrm{r}}^{4} fie_{\mathrm{r}}}{3600\rho \mathrm{v}} + x_{r}^{4} \mathrm{D}_{\mathrm{r}}^{4} \mathrm{CU_p} + \\ \mathrm{p} \sum_{\mathrm{r}=1}^{\mathrm{R}} \frac{\mathrm{D}_{\mathrm{r}}^{5} fse_{\mathrm{i}}}{3600\rho \mathrm{v}} + x_{\mathrm{i}}^{5} \mathrm{D}_{\mathrm{i}}^{5} \mathrm{CU_p} \end{array} \right\} \tag{3}$$

For the regeneration cost, the capital and the operating costs of the property interceptors are calculated as follows,

$$RC = \mathrm{K_F} \sum_{\mathrm{r}\in\mathrm{R}} \mathrm{CU_r} FR_{\mathrm{r}}^{\alpha} + \mathrm{H_Y} \sum_{\mathrm{r}\in\mathrm{R}} \mathrm{CUM_r} FR_{\mathrm{r}} \tag{4}$$

Here, $\mathrm{H_Y}$ represents the plant operating hours per year, $\mathrm{CU_w}$ is the fresh water unit cost, $fws_{\mathrm{w,j}}$ is the flow rate of the fresh water *w* in the process sink *j*, $\mathrm{K_F}$ is the factor used to annualize the capital costs, $\mathrm{CU_r}$ is the investment cost coefficient, α is an exponent for the cost function, $\mathrm{CUM_r}$ is the unit cost for the removed pollutants in each interceptor, $\mathrm{D}_{\mathrm{i,j}}^{1}$ is the distance between source *i* and sink *j*, $\mathrm{D}_{\mathrm{i,r}}^{2}$ is the distance between source *i* and interceptor *r*, $\mathrm{D}_{\mathrm{r,j}}^{3}$ is the distance between interceptor *r* and process sink *j*, $\mathrm{D}_{\mathrm{r}}^{4}$ is the distance between the interceptor *r* and the environmental discharge, $\mathrm{D}_{\mathrm{i}}^{5}$ is the distance between the source *i* and the environmental discharge, ρ is the water density, v is the velocity, p is a parameter for cross-plant pipeline capital cost, and $\mathrm{CU_p}$ is the pipe unit cost. In addition, the value of α usually takes values between 0.6 and 0.8; therefore, Eq. 4 is nonlinear and nonconvex.

Mass balance for each process source

The flow rate of each process source ($\mathrm{FS_i}$) can be split and directed to each process sink ($fss_{\mathrm{i,j}}$), to each interceptor ($fsi_{\mathrm{i,r}}$), and to the environmental discharge (fse_{i}),

$$\mathrm{FS_i} = \sum_{\mathrm{j}\in\mathrm{J}} fss_{\mathrm{i,j}} + \sum_{\mathrm{r}\in\mathrm{R}} fsi_{\mathrm{i,r}} + fse_{\mathrm{i}}, \qquad i \in I \tag{5}$$

Notice that the above equation represents the reuse and recycle of the water sources in the same plant and other plants as well as in the interceptors, which are shared by the participating plants. In addition, it considers the wastewater flow rate discharged to the environment.

Mass balances and property mixing rules for each process sink

The flow rate required by the process sinks ($\mathrm{FU_j}$) is generated by the portions of process sources sent to them ($fss_{\mathrm{i,j}}$), the flow rates from the interceptors ($fis_{\mathrm{r,j}}$), and the fresh water ($fws_{\mathrm{w,j}}$) needed to meet the constraints respect to lower and upper limits ($\psi_{\mathrm{p}}(\mathrm{Pu}_{\mathrm{p,j}}^{\max})$, $\psi_{\mathrm{p}}(\mathrm{Pu}_{\mathrm{p,j}}^{\min})$) given in terms of properties (Notice that to avoid numerical complications due to the nonlinear mixing relationships, the property operator is the optimization variable instead of the properties). For overcoming the difficulty of quantifying the properties as function of composition due to the large set of components usually present in the waste process streams, Shelley and El-Halwagi[29] introduced a novel design paradigm called component-less design. They proposed mixing rules for the properties similar to the component balances using the property operators shown in Table 1. Therefore, the mass and property balances for the mixers before the process sinks in each plant are stated as follows:

$$\mathrm{FU_j} = \sum_{\mathrm{r}\in\mathrm{R}} fis_{\mathrm{r,j}} + \sum_{\mathrm{i}\in\mathrm{I}} fss_{\mathrm{i,j}} + \sum_{\mathrm{w}\in\mathrm{W}} fws_{\mathrm{w,j}}, \qquad j \in J \tag{6}$$

$$\sum_{\mathrm{r}\in\mathrm{R}} \left[\psi_{\mathrm{p}} \left(Pi_{\mathrm{p,r}}^{\mathrm{out}} \right) fis_{\mathrm{r,j}} \right] + \sum_{\mathrm{i}\in\mathrm{I}} \left[\psi_{\mathrm{p}} \left(\mathrm{Ps_{p,i}} \right) fss_{\mathrm{i,j}} \right] + \sum_{\mathrm{w}\in\mathrm{W}} \left[\psi_{\mathrm{p}} \left(\mathrm{Pw_{p,w}} \right) fws_{\mathrm{w,j}} \right] \le \psi_{\mathrm{p}} \left(\mathrm{Pu}_{\mathrm{p,j}}^{\max} \right) \mathrm{FU_j}, \qquad j \in J; p \in P \tag{7}$$

$$\sum_{\mathrm{r}\in\mathrm{R}} \left[\psi_{\mathrm{p}} \left(Pi_{\mathrm{p,r}}^{\mathrm{out}} \right) fis_{\mathrm{r,j}} \right] + \sum_{\mathrm{i}\in\mathrm{I}} \left[\psi_{\mathrm{p}} \left(\mathrm{Ps_{p,i}} \right) fss_{\mathrm{i,j}} \right] + \sum_{\mathrm{w}\in\mathrm{W}} \left[\psi_{\mathrm{p}} \left(\mathrm{Pw_{p,w}} \right) fws_{\mathrm{w,j}} \right] \ge \psi_{\mathrm{p}} \left(\mathrm{Pu}_{\mathrm{p,j}}^{\min} \right) \mathrm{FU_j}, \qquad j \in J; p \in P \tag{8}$$

Here, $\psi_{\mathrm{p}}(\mathrm{Ps_{p,i}})$ and $\psi_{\mathrm{p}}(\mathrm{Pw_{p,w}})$ are the property operators for each process source *i* and each fresh water *w*, respectively. Notice that the outlet property operator for each interceptor ($\psi_{\mathrm{p}}(Pi_{\mathrm{p,r}}^{\mathrm{out}})$) and the flow rates that are sent to each sink are optimization variables; therefore, the term $\psi_{\mathrm{p}}(Pi_{\mathrm{p,r}}^{\mathrm{out}}) fis_{\mathrm{r,j}}$ is bilinear and, consequently, Eqs. 7 and 8 are nonlinear and nonconvex relationships.

Mass balances and property mixing rules for the property interceptors

A set of shared property interceptors are available to satisfy the property constraints for the sinks in the participating plants and for the wastewater discharged to the environment. The conditions in terms of the flow rate and property operators at the inlet to each interceptor (FR_r, $\psi_p(Pi^{in}_{p,r})$) are determined by the flow rates that are sent from the process sources ($fsi_{i,r}$) and from other interceptors ($fii_{r_1,r}$),

$$FR_r = \sum_{i\in I} fsi_{i,r} + \sum_{\substack{r_1\in R \\ r_1\neq r}} fii_{r_1,r}, \qquad r \in R \tag{9}$$

$$\psi_p\left(Pi^{in}_{p,r}\right)FR_r = \sum_{i\in I}\left[\psi_p\left(Ps_{p,i}\right)fsi_{i,r}\right] + \sum_{\substack{r_1\in R \\ r_1\neq r}}\left[\psi_p\left(Pi^{out}_{p,r_1}\right)fii_{r_1,r}\right], \qquad r \in R; p \in P \tag{10}$$

Because the inlet and outlet conditions for the interceptors and the flow rates between them are unknown variables, then Eq. 10 is a nonlinear and nonconvex expression. Finally, the flow rate from each interceptor can be segregated and sent to the process sinks ($fis_{r,j}$), the environmental discharge (fie_r), and other interceptors (fii_{r,r_1}). In addition, the outlet property operators are determined by the efficiency factor for each interceptor ($RR_{r\psi_p(Pp)}$),

$$FR_r = \sum_{j\in J} fis_{r,j} + \sum_{\substack{r_1\in R \\ r_1\neq r}} fii_{r,r_1} + fie_r, \qquad r \in R \tag{11}$$

$$\psi_p\left(Pi^{out}_{p,r}\right) = \psi_p\left(Pi^{in}_{p,r}\right)RR_{r,\psi_p\left(P_p\right)} \tag{12}$$

The efficiency factor depends on the type, configuration, and design variables of the treatment unit *r,* and it is determined before the optimization process through simulations or empirically to avoid additional numerical complexities. In this regard, in this work, like all previous works for interplant water integration, the efficiency factors are assumed as constants to avoid additional numerical complications. This assumption works properly because several interceptors with a given configuration and operating conditions can be simulated before optimization providing a good correlation for their efficiency, whereas the associated costs depend only on the treated flow rate. To account for the limits on the flow rate and properties to get a proper function for the treatment units, the model can include property and flow rate constraints for the inlet flow rate to each treatment unit. This way, the optimization model must select the interceptor to use and the flow rate treated. In addition, the efficiency factor could be positive or negative depending on the type of interceptor and the property treated.

Mass balances and property mixing rules for the mixer before the waste stream discharged to the environment

The flow rate discharged to the environment (*FE*) and the value for the property operators ($\psi_p(Pe^{max}_p)$, $\psi_p(Pe^{min}_p)$) in the waste stream discharged to the environment is formed by the portions of the flow rates of process sources (fse_i) and the property interceptors (fie_r). The last term is a variable that multiplies the outlet property operator of the interceptors, which yields a bilinear term in Eqs. 14 and 15.

$$FE = \sum_{i\in I} fse_i + \sum_{r\in R} fie_r \tag{13}$$

$$\sum_{i\in I}\left[\psi_p\left(Ps_{p,i}\right)fse_i\right] + \sum_{r\in R}\left[\psi_p\left(Pi^{out}_{p,r}\right)fie_r\right] \le \psi_p\left(Pe^{max}_p\right)FE, \qquad p \in P \tag{14}$$

$$\sum_{i\in I}\left[\psi_p\left(Ps_{p,i}\right)fse_i\right] + \sum_{r\in R}\left[\psi_p\left(Pi^{out}_{p,r}\right)fie_r\right] \ge \psi_p\left(Pe^{min}_p\right)FE, \qquad p \in P \tag{15}$$

Determination of pipe segments

To account for the unit cost of the pipe segments required in the optimal configuration, the following general disjunction is included to activate the binary variables x associated to the existence of any pipe segment in the superstructure,

$$\begin{bmatrix} X_{m,n} \\ f_{m,n} \ge M^{min}_{m,n} \\ f_{m,n} \le M^{max}_{m,n} \end{bmatrix} \vee \begin{bmatrix} \neg X_{m,n} \\ f_{m,n} = 0 \end{bmatrix}$$

Here, $X_{m,n}$ is a Boolean variable for selecting the unit cost of any pipe, $f_{m,n}$ is the flow rate in any pipe segment that begins at m and finishes in n; therefore, m could be a process source or a treatment unit, while n could be a process sink, a treatment unit, or the waste that is discharged to environment. Furthermore, $M^{min}_{m,n}$ and $M^{max}_{m,n}$ are lower and upper limits for the flow rate to determine the existence of any pipe segment. To reformulate the above disjunction as algebraic expressions, the big-M technique is used (Nemhauser and Wolsey[35]). Then, the binary variables are activated when a pipe segment is required in the EIP. Hence, if the Boolean variable $X_{m,n}$ is true, then its corresponding binary variable $x_{m,n}$ is equal to one, and the flow rate in the pipe segment must be bigger than a lower limit ($M^{min}_{m,n}$) and lower than an upper limit ($M^{max}_{m,n}$). On the other hand, if the Boolean variable $X_{m,n}$ is false, its corresponding binary variable $x_{m,n}$ is equal to zero; in this case, the flow rate in the pipe segment is equal to zero, too. This situation is modeled through the following relationships:

$$f_{m,n} \ge x_{m,n} M^{min}_{m,n}, \qquad m \in M;\ n \in N \tag{16}$$

$$f_{m,n} \le x_{m,n} M^{max}_{m,n}, \qquad m \in M;\ n \in N \tag{17}$$

Notice that the binary variable $x_{m,n}$ is a general representation of the binary variables ($x^1_{i,j}$, $x^2_{i,r}$, $x^3_{r,j}$, x^4_r and x^5_i) used to determine any pipe segment in the superstructure and its associated cost (see Eq. 3).

Model Reformulations

Notice that the model previously presented is a mixed integer nonlinear programming problem and it is nonconvex because it has four bilinear terms in Eqs. 7, 8, 10, 14, and 15, and an exponential term in Eq. 4. Solving this type of problems is quite complicated (see, e.g., Rubio-Castro et al.;[20,21] Teles et al.[36]). In this regard, for any bilinear term between the continuous variables x and y, there are infinite combinations for upper and lower bounds of them as it can be seen in Figure 2a. Then, if either x or y is transformed into a known discrete parameter, the bilinear term can be

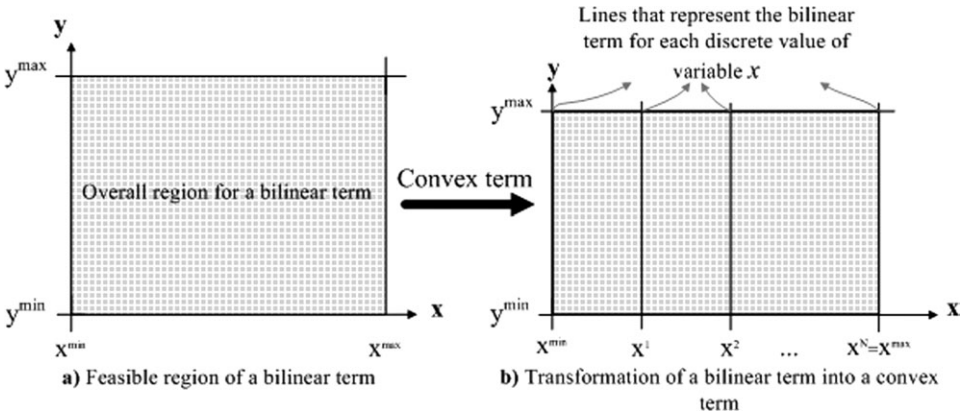

Figure 2. Bilinear term and its transformation into a convex term.

reformulated as a convex linear term, but there are infinite combinations between each discrete value and any value of the continuous variable as it can be seen in Figure 2b, which shows the feasible lines for a bilinear term by discretizing the continuous variable x.

It should be noted in Figure 2b that to cover all the possible solutions using the discrete values to guarantee the global optimal solution, it is required to discretize the continuous variable x into an infinite number of points, which is a task impossible to do. Therefore, for getting a global optimal solution, it is required the construction of a convex underestimating problem for comparing iteratively a mixed-integer linear problem (upper bound) and an under estimator relaxed problem (lower bound) until reaching a minimum gap between them as it is shown in Figure 3. The general spatial search shown in Figure 3 has been the central theme of many methodologies previously developed; a complete overview about the existing formulations for piecewise mixed integer linear programming (MILP) underestimators and overestimators for bilinear programs was presented by Wicaksono and Karimi.[37] Among others, the GOP algorithm (see Visweswaran and Floudas[38]) and the branch and bound algorithm (see Al-Khayyal and Falk[39]) are formulations where a good lower bound is the key factor to get the global optimal solution. There is a huge feasible region for finding a valid lower bound as it can be seen in Figure 4a, which shows a nonconvex function respect to any variables x and y, and its corresponding region where a valid lower bound should be found to compare with the upper bound. Therefore, some works have focused on improving the lower bounds, for example, Maranas and Floudas[40,41], Androulakis et al.,[42] Liu and Floudas,[43] Adjiman et al.[44] and Karuppiah and Grossmann.[45]

The article by Karuppiah and Grossmann[45] proposed a piecewise under and overestimators to approximate the nonconvex bilinear terms to obtain a convex relaxation, whose solution gives a lower bound on the global optimum, and certain heuristics are followed as branching rules. Therefore, the general approach of Karuppiah and Grossmann[45] is the basis of the proposed procedure in this work; however, in this work, the partition is done on the property operator (none on the flow rates). Although this choice increases the problem size, its advantage is that every bilinear term depends on the conditions (flow rates and property operator) in the interceptors. Consequently, if the property operators at the inlet to the interceptors are transformed into parameters, their effect on the overall process can be determined varying only the value of the property operator in the regeneration zone, which is easier than to analyze the effect of all possible flow rates on the superstructure.

In this way, this article proposes a new strategy for the branching variables, which is based on the property operators and the goal of this contract algorithm is to reduce the original feasible region for the lower bound and to approximate it to the upper bound. Either variable x or y is discretized in the valid range to transform the original feasible region in several feasible regions for the different intervals of the discretized variable. This approach allows to find at least one lower bound in some interval of the discretized variable, which is better than the original lower bound; for example, Figure 4b shows the contraction of the original feasible region and it can generate better lower bounds (i.e., notice in Figure 4b that the original gap β is bigger than β^1, β^2, and β^3). In addition, with this technique, all regions of bilinear terms are covered without considering an infinite number of discretized values for the variables. However, the best lower bound so obtained may be unfeasible and the evolution of the spatial branch and bound procedure depends strongly on the branching variables. For example, if the branching variable is only the variable x, the area for getting a valid lower bound is the whole area under the curve corresponding to the upper bound in each interval of x (see Figure 5a). On the other hand, if the branching variables are both x and y, discretizing the first one and using the second one as lower bound and upper bound for each interval of x, each feasible region of Figure 5a is transformed into two feasible regions as it is presented in Figure 5b. In this way, the feasible region contraction improves the lower bounds respect to the

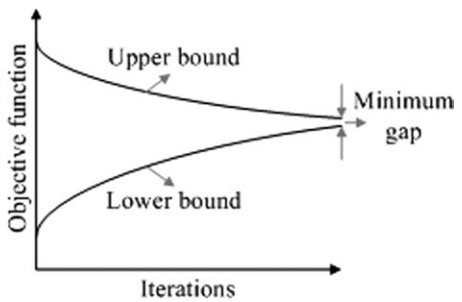

Figure 3. Evolution of a spatial branch and bound procedure.

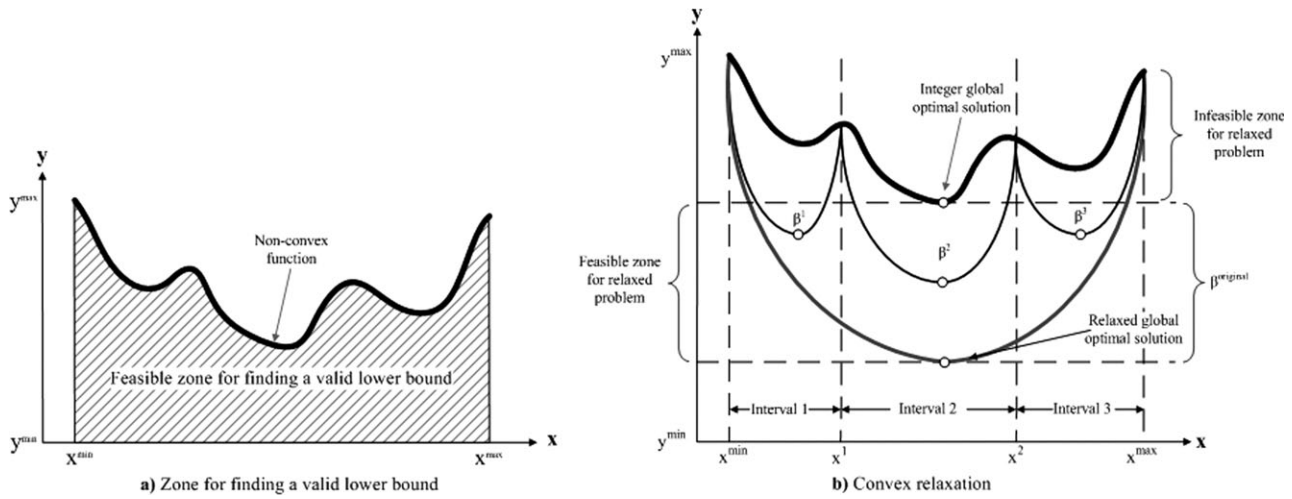

Figure 4. Lower bound of a nonconvex function.

original and actual lower bounds when only one variable is used for the branching variables due to the feasible region was reduced. In other words, now there are two values for the lower bound in each interval of x for selecting the one that represents the best valid lower bound. These two values are estimated like the tree shown in Figure 5c. In addition, y^1, y^2, y^3 represent the optimized values of y for each interval of x; in this regard, notice that only the variable x was discretized into known parameters and the variable y is a continuous variable yielding convex linear relationships.

The reformulation and linearization for the upper and lower bounds as well as the objective function that are used during the proposed branch and bound procedure are presented as follows.

Reformulation and linearization (upper bound)

To get a good upper bound, the mixed integer nonlinear programming problem is reformulated as a MILP problem by discretizing the property operators of the streams into a finite number of intervals (n^1) via a convex hull reformulation (Raman and Grossmann[46]). A good upper bound helps significantly to eliminate sections where the optimal solution is not found since the early stages of the optimization approach. Although this discrete reformulation can consume additional CPU time, the reduction so achieved in the

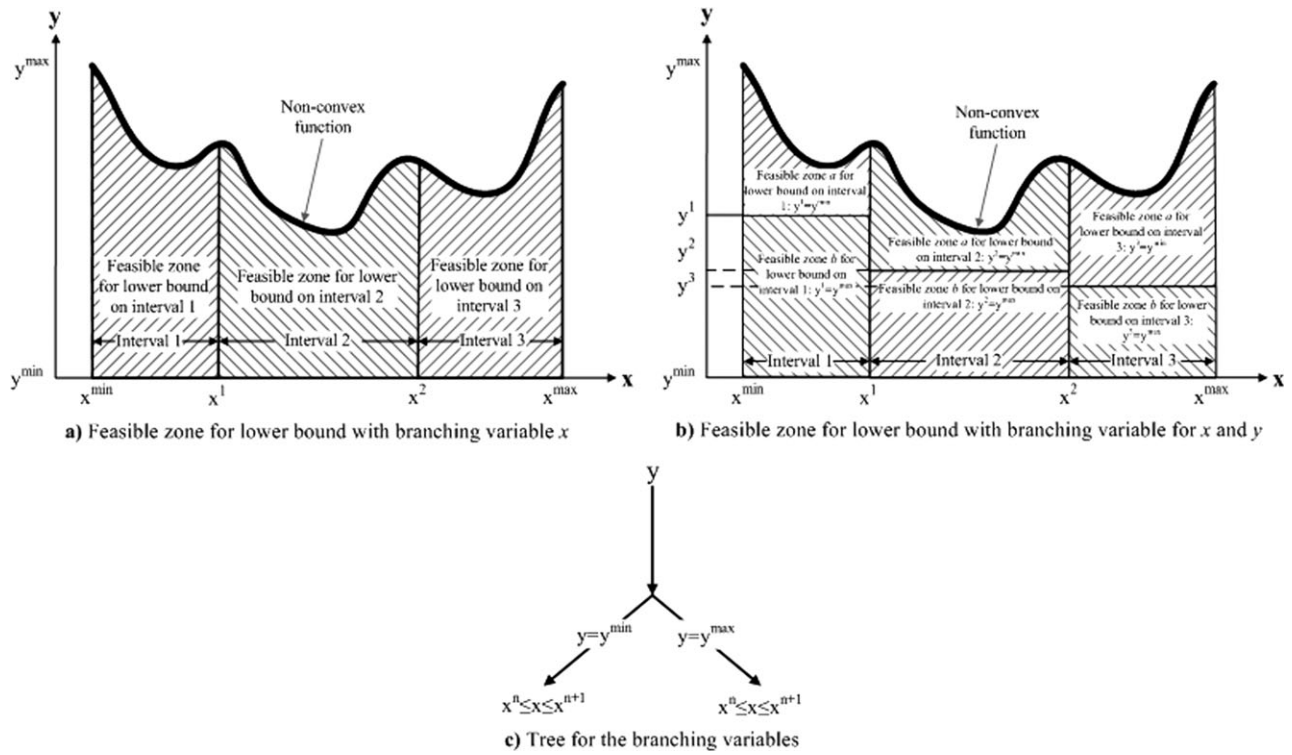

Figure 5. Effect of the branching variables over the lower bound.

solution space improves significantly the overall CPU time. As shown by Pham et al.[32] and Rubio-Castro et al.,[20,21] these discrete reformulations can yield solutions near the optimal one. Therefore, using this discretized formulation for the upper bound, the root node of the upper bound can be near the optimal eliminating unfavorable sections and reducing the overall consumed CPU time. This section presents an explanation for the discretization of any bilinear term yx (here y is any flow rate and x is any property operator).

First, each bilinear term is replaced by a new variable ($B^1_{k,p}$),

$$B^1_{k,p} = F^1_k \psi^1_{k,p}, \qquad k \in K, p \in P \tag{18}$$

Here, F^1_k is any flow rate that appears in a bilinear term like FR_r, $fii_{r_1,r}$, $fis_{r,j}$, and fie_r; and $\psi^1_{k,p}$ is any property operator that appears in a bilinear term which is transformed into a discrete value $\gamma^1_{k,p,q}$ for transforming the bilinear term into a convex term using the following equation (see Pham et al.[32]).

$$\gamma^1_{k,p,q^1} = \min\left(\psi^1_{k,p}\right) + \left(q^1 - 1\right)\frac{\max\left(\psi^1_{k,p}\right) - \min\left(\psi^1_{k,p}\right)}{n^1}, \quad k \in K, p \in P, q^1 \in Q^1 \tag{19}$$

where n^1 represents the number of intervals used to split $\psi^1_{k,p}$. Then, the optimization problem consists in finding the optimal discrete value $\gamma^1_{k,p,q}$. This situation is modeled through the next disjunction:

$$\begin{matrix} \vee \\ q^1 = 1 \ldots n^1 + 1 \end{matrix} \begin{bmatrix} Y^1_{k,p,q^1} \\ B^1_{k,p} = F^1_k \gamma^1_{k,p,q^1} \end{bmatrix}, \quad k \in K; p \in P$$

This disjunction implies that only one q^1 (discrete choice) must be selected for the bilinear terms. Thus, when the Boolean variable $Y^1_{k,p,q1}$ is true, its bilinear term $B^1_{k,p}$ must be equal to $F^1_k \gamma^1_{k,p,q1}$ (notice that $\gamma^1_{k,p,q1}$ is constant), and the other Boolean variables are false so their corresponding $B^1_{k,p}$ are equal to zero. This disjunction is modeled using the convex hull reformulation. First, the optimization variables are disaggregated as follows,

$$B^1_{k,p} = \sum_{q^1 \in Q^1} \beta^1_{k,p,q^1}, \qquad k \in K, p \in P \tag{20}$$

$$F^1_k = \sum_{p \in P} \sum_{q^1 \in Q^1} \delta^1_{k,p,q^1}, \qquad k \in K \tag{21}$$

Notice in previous equation that the variable F^1_k is valid for the domain k, while the corresponding disaggregated variable is valid for the domains k, p, and q^1; therefore, to yield proper relationships valid for the same domain (i.e., k) the summation of the disaggregated variable is over p and q^1.

Then, the equation is reformulated in terms of the disaggregated variables ($\beta^1_{k,p,q1}$, $\delta^1_{k,p,q}$),

$$\beta^1_{k,p,q^1} = \delta^1_{k,p,q} \gamma^1_{k,p,q^1}, \qquad k \in K; p \in P; q^1 \in Q^1 \tag{22}$$

Upper limits ($M^{max}_{B^1_{k,p}}, M^{max}_{F_k}$) are established for the disaggregated variables as follows,

$$\beta^1_{k,p,q^1} \le M^{max}_{B^1_{k,p}} y^1_{k,p,q^1}, \qquad k \in K; p \in P; q^1 \in Q^1 \tag{23}$$

$$\delta^1_{k,p,q^1} \le M^{max}_{F_k} y^1_{k,p,q^1}, \qquad k \in K; p \in P; q^1 \in Q^1 \tag{24}$$

The following equation is used to select the active region through one binary variable and to select only one discrete value for each property operator,

$$\sum_{k \in K} \sum_{q^1 \in Q^1} y^1_{k,p,q^1} = 1, \qquad p \in P \tag{25}$$

Notice that previous equations are valid for the domain of p; therefore, the summation of the binary variables $y^1_{k,p,q1}$ is for the domains k and q^1 to yield the same validity for the domain p, and the next equation allows knowing the value for the property operator,

$$\psi^1_{k,p} = \sum_{q^1 \in Q^1} \gamma^1_{k,p,q^1} y^1_{k,p,q^1}, \qquad k \in K; p \in P \tag{26}$$

Reformulation and linearization (lower bound)

Piecewise under estimators based on the properties are calculated to obtain valid lower bounds. First, every bilinear term is replaced by a new variable ($B^2_{k,p}$),

$$B^2_{k,p} = F^2_k \psi^2_{k,p}, \qquad k \in K; p \in P \tag{27}$$

Both the flow rate (F^2_k) and the property operator ($\psi^2_{k,p}$) are limited by lower and upper limits ($F^{2,min}_k$, $F^{2,max}_k$, $\psi^{2,max}_{k,p}$, $\psi^{2,max}_{k,p}$),

$$F^{2,min}_k \le F^2_k \le F^{2,max}_k, \qquad k \in K \tag{28}$$

$$\psi^{2,min}_{k,p} \le \psi^2_{k,p} \le \psi^{2,max}_{k,p}, \qquad k \in K; p \in P \tag{29}$$

Using these limits, the following linear constraints are obtained, which are the convex and concave envelopes of the bilinear terms over the given bounds (Sherali and Alameddine[47]),

$$B^2_{k,p} \ge F^{2,min}_k \psi^2_{k,p} + \psi^{2,min}_{k,p} F^2_k - F^{2,min}_k \psi^{2,min}_{k,p}, \quad k \in K; p \in P \tag{30}$$

$$B^2_{k,p} \ge F^{2,max}_k \psi^2_{k,p} + \psi^{2,max}_{k,p} F^2_k - F^{2,max}_k \psi^{2,max}_{k,p}, \quad k \in K; p \in P \tag{31}$$

$$B^2_{k,p} \le F^{2,min}_k \psi^2_{k,p} + \psi^{2,max}_{k,p} F^2_k - F^{2,min}_k \psi^{2,max}_{k,p}, \quad k \in K; p \in P \tag{32}$$

$$B^2_{k,p} \le F^{2,max}_k \psi^2_{k,p} + \psi^{2,min}_{k,p} F^2_k - F^{2,max}_k \psi^{2,min}_{k,p}, \quad k \in K; p \in P \tag{33}$$

To reduce the gap between the upper and lower bounds and to reduce the number of iterations required by the branch and bound procedure, one partition over the original domain for the property operator $D_{k,p} = [\psi^{2,min}_{k,p}, \psi^{2,max}_{k,p}]$ is made using the points $\psi^{min}_{k,p} = \gamma^2_{k,p,1}, \gamma^2_{k,p,2}, \ldots, \gamma^2_{k,p,N+1} = \psi^{2,max}_{k,p}$ with piecewise linear under estimators for the bilinear terms in each partition (see, e.g., Karuppiah and Grossmann[45]). Then, the construction of the piecewise under estimators can be set using the following disjunction,

$$\bigvee_{q^2=1\ldots NQ^2}\begin{bmatrix} Y^2_{k,p,q^2} \\ B^2_{k,p} \geq F^{2,min}_k \psi^2_{k,p} + \gamma^2_{k,p,q^2} F^2_k - F^{2,min}_k \gamma^2_{k,p,q^2} \\ B^2_{k,p} \geq F^{2,max}_k \psi^2_{k,p} + \gamma^2_{k,p,q^2+1} F^2_k - F^{2,max}_k \gamma^2_{k,p,q^2+1} \\ B^2_{k,p} \leq F^{2,min}_k \psi^2_{k,p} + \gamma^2_{k,p,q^2+1} F^2_k - F^{2,min}_k \gamma^2_{k,p,q^2+1} \\ B^2_{k,p} \leq F^{2,max}_k \psi^2_{k,p} + \gamma^2_{k,p,q^2} F^2_k - F^{2,max}_k \gamma^2_{k,p,q^2} \end{bmatrix}, \quad k \in K; p \in P$$

where $Y^2_{k,p,q2}$ is a Boolean variable used to select the optimal value for the partitions and $Y^2_{k,p,q2}$ is a parameter known that represents the partitions of the original domain of the property operator, which is calculated as follows:

$$\gamma^2_{k,p,q^2} = \min\left(\psi^2_{k,p}\right) + \left(q^2 - 1\right)\frac{\max\left(\psi^2_{k,p}\right) - \min\left(\psi^2_{k,p}\right)}{n^2}, \quad k \in K; p \in P; q^2 \in Q^2 \quad (34)$$

Here, n^2 is the number of partitions of the original domain of property operator ($\psi^2_{k,p}$). To reformulate the last disjunction, the convex hull technique is used, which includes a binary constraint for activating just one region between the partitions,

$$\sum_{q^2 \in Q^2} y^2_{k,p,q^2} = 1, \quad k \in K; p \in P \quad (35)$$

The flow rate, the property operator, and the bilinear term must be expressed in terms of their disaggregated variables ($\delta^2_{k,p,q2}$, $\pi^2_{k,p,q2}$, $\beta^2_{k,p,q2}$),

$$F^2_k = \sum_{q^2 \in Q^2}\sum_{p \in P} \delta^2_{k,p,q^2}, \quad k \in K \quad (36)$$

Notice in previous equation that the summation on the right hand side is for the indexes q^2 and p to yield the validity for the index p on both sides of the relationship.

$$\psi^2_{k,p} = \sum_{q^2 \in Q^2} \pi^2_{k,p,q^2}, \quad k \in K; p \in P \quad (37)$$

$$B^2_{k,p} = \sum_{q^2 \in Q^2} F^{2,min}_k \pi^2_{k,p,q^2} + \gamma^2_{k,p,q^2}\delta^2_{k,p,q^2} - F^{2,min}_k \gamma^2_{k,p,q^2} y^2_{k,p,q^2}, \quad k \in K; p \in P \quad (38)$$

$$B^2_{k,p} = \sum_{q^2 \in Q^2} F^{2,max}_k \pi^2_{k,p,q^2} + \gamma^2_{k,p,q^2+1}\delta^2_{k,p,q^2} - F^{2,max}_k \gamma^2_{k,p,q^2+1} y^2_{k,p,q^2}, \quad k \in K; p \in P \quad (39)$$

$$B^2_{k,p} = \sum_{q^2 \in Q^2} F^{2,min}_k \pi^2_{k,p,q^2} + \gamma^2_{k,p,q^2+1}\delta^2_{k,p,q^2} - F^{2,min}_k \gamma^2_{k,p,q^2+1} y^2_{k,p,q^2}, \quad k \in K; p \in P \quad (40)$$

$$B^2_{k,p} = \sum_{q^2 \in Q^2} F^{2,max}_k \pi^2_{k,p,q^2} + \gamma^2_{k,p,q^2}\delta^2_{k,p,q^2} - F^{2,max}_k \gamma^2_{k,p,q^2} y^2_{k,p,q^2}, \quad k \in K; p \in P \quad (41)$$

and the disaggregated variables are restricted by the following limits,

$$\gamma^2_{k,p,q^2+1} y^2_{k,p,q^2} \leq \pi^2_{k,p,q^2} \leq \gamma^2_{k,p,q^2} y^2_{k,p,q^2}, \; k \in K; p \in P; q^2 \in Q^2 \quad (42)$$

$$F^{2,max}_k y^2_{k,p,q^2} \leq \delta^2_{k,p,q^2} \leq F^{2,min}_k y^2_{k,p,q^2}, \quad k \in K; p \in P; q^2 \in Q^2 \quad (43)$$

Linearization of the objective function

Finally, the nonlinear term in Eq. 4 was linearized through the strategy used by Rubio-Castro et al.[20]. In this work, a value of 0.7 for α is used, and the piecewise linearization procedure for a range of flow rate from 0 to 300,000 is shown in Figure 6; in this figure, the linear regression was made in the following segments for the flow rate: 0–5000, 5000–50,000, 50,000–100,000, and 100,000–300,000. Notice that just one segment must be selected; therefore, a disjunction and its corresponding convex hull reformulation are used to model this situation, where the term FR^{α}_r that appears in Eq. 4 is replaced by B^3_r,

$$\bigvee_{z \in Z}\begin{bmatrix} Y^3_{r,z} \\ B^3_r = A_{r,z} FR_r + C_{r,z} \end{bmatrix}, \quad r \in R$$

$$B^3_r = \sum_{z \in Z} \beta^3_{r,z}, \quad r \in R \quad (44)$$

$$FR_r = \sum_{z \in Z} \delta^3_{r,z}, \quad r \in R \quad (45)$$

$$\beta^3_{r,z} = \delta^3_{r,z} A_{r,z} + C_{r,z} y^3_{r,z}, \quad r \in R; z \in Z \quad (46)$$

$$\beta^3_{r,z} \leq M^{max}_{B^3_r} y^3_{r,z}, \quad r \in R; z \in Z \quad (47)$$

$$\delta^3_{r,z} \leq M^{max}_{FR_{r,z}} y^3_{r,z}, \quad r \in R; z \in Z \quad (48)$$

$$\delta^3_{r,z} \geq M^{min}_{FR_{r,z}} y^3_{r,z}, \quad r \in R; z \in Z \quad (49)$$

$$\sum_{z \in Z} y^3_{r,z} = 1, \quad r \in R \quad (50)$$

In the above equations, the subscript z is used to denote a linear segment of the original curve, $\beta^3_{r,z}$ is the disaggregated bilinear term for B^3_r, $\delta^3_{r,z}$ is the disaggregated flow rate for FR_r, $M^{max}_{B^3_r}$ is the upper limit for the bilinear term $\beta^3_{r,z}$, $M^{max}_{FRr,z}$ and $M^{min}_{FRr,z}$ are the upper and lower limits for $\delta^3_{r,z}$, and $y^3_{r,z}$ is a binary variable used to select the optimal value of the slope

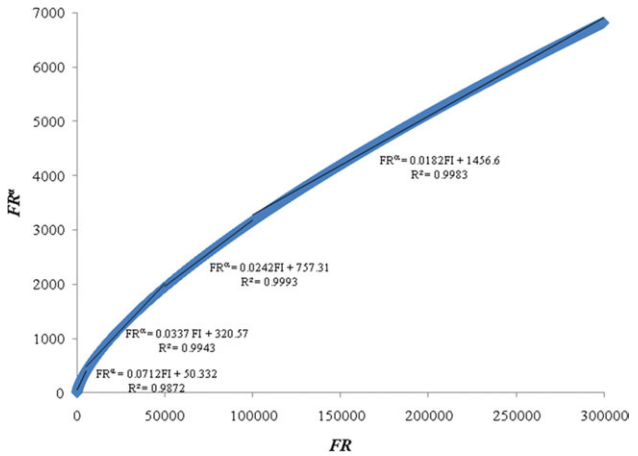

Figure 6. Linearization of the exponential term for the capital cost of treatment units.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com]

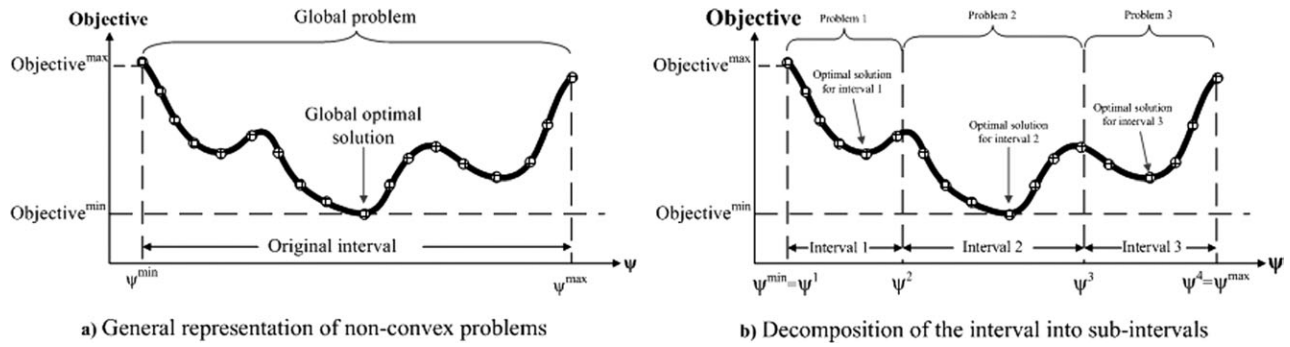

Figure 7. Partitions of the original interval of a nonconvex problem.

$A_{r,z}$ and the intersection $C_{r,z}$ for each line segment. Figure 6 shows the data for A_z and C_z for a value of 0.7 for α, and the values for $M^{min}_{FRr,z}$ and $M^{max}_{FRr,z}$.

Global Optimization Algorithm

To explain the proposed spatial branch and bound algorithm, Figure 7a is the starting point. This figure shows a simple representation of a nonconvex objective function respect to a discretized property operator. Notice that each point of the objective function corresponds to a specific value for the discretized property operator. Hence, for getting the global optimal solution, it is required an infinite number of discretized values, which is unreasonable. Then, a big number of discrete points is a good approximation; however, it increases the problem size and demands a big computational effort. Therefore, a decomposition of the original interval into subintervals, as it is shown in Figure 7b, reduces the problem size. Then, the new problems are solved and their solutions are compared to select the best solution (which represents a valid upper value for the objective function). Obviously, when the problem has several properties, it must take into account all possible combinations between the new intervals, for covering the overall search space.

To ensure that the best upper bound solution obtained with the above strategy corresponds to the global optimal solution, it is required to determine the corresponding lower convex under estimator and to implement a spatial branch and bound procedure as it is shown in Figure 8a. This figure shows a nonconvex objective function and its corresponding convex relaxation. Points A_1 and A_1' are the optimal solutions for integer (upper bound) and relaxed (lower bound) problems, respectively; there is a gap (β) that should be reduced to the minimum value to ensure getting the global optimal solution. In this regard, there are three routes (left, center, and right) to get the point A_1' for the minimal gap of the goal line (dashed line), as it can be seen in Figure 8b. Then, when the original interval is divided into subintervals as it was explained above, there are two subproblems for the upper and lower bounds for each subsection, as it is shown in Figure 8b.

Finally, the spatial branch and bound consists in solving both the integer and relaxed problems iteratively for each subinterval. The results obtained at iteration t for the property operator and flow rate at the interceptor of the integer problem (upper bound) are used to determine two new subintervals for the iteration $t + 1$, one on the left and the other one on the right hand sides of the property operator, which should be solved using the flow rates inlet to the interceptors like lower and upper bounds (see Figure 5c). To select the best solution for all intervals, the lowest solution for the integer problem and the highest solution for the relaxed problem are compared; the search stops when the gap between them is less or equal than a given tolerance. For example, considering that the gray points in Figure 8b are the optimal solutions in the first iteration for each interval, then the best solutions for the integer and relaxed problems are the gray points in the Intervals 2 and 1, respectively. For the second iteration, a partition is made (dashed line: —·—·—) in each gray point of the integer solution for generating two new intervals (one on the left and the other on the right hand side) for the integer and relaxed problems; notice that in this iteration the best solution for the integer problem is represented by the white point located in the Interval 2 right, while for the relaxed problem the best solution is in the Interval 1 left. Also, the white points in the Intervals 1 right, 3 right, and 3 left represent the relaxed values bigger than the best integer solution found; therefore, these points are in unfavorable sections for the relaxed problem so they must be

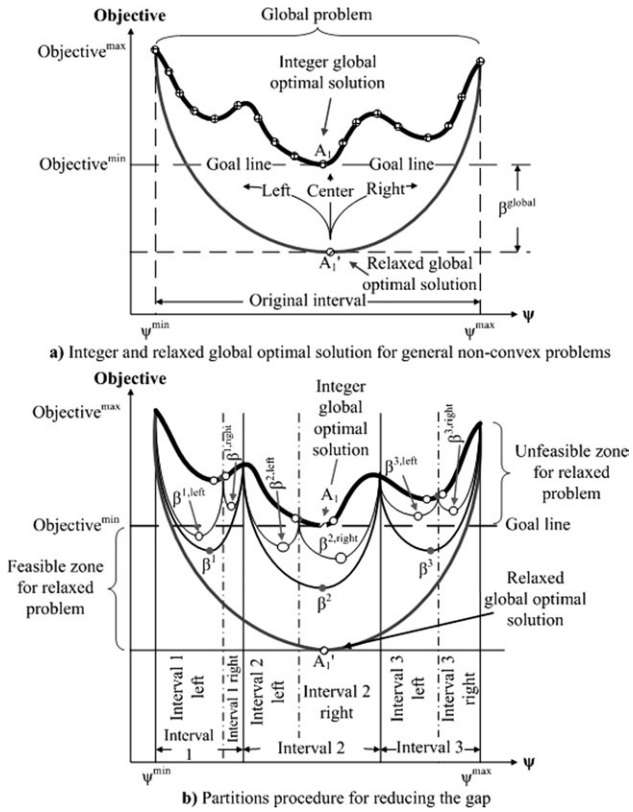

Figure 8. Branching variables procedure.

eliminated. As a result, they will not be considered in the following iterations. In addition, the solution for the integer problem could fall in the extreme of its corresponding interval. In this case, if such solution represents the global optimal solution for this interval, in the next iteration, the integer solution will be the same and the relaxed solution corresponding to this interval should be used for the next iterations for comparing it with the integer solution of the remain intervals.

Here, the upper bound is selected for doing the partitions trough the iterations to evaluate the relaxed solution of each integer problem during the proposed algorithm. Then, once the minimum gap is reached, the corresponding upper bound represents the global optimal solution. Furthermore, the partition schemes (both the upper bound as well as the lower bound) are similar (see Eqs. 19 and 34), but for the lower bound the optimal value of each bilinear term can be selected from the points in the extreme of the partitions or inside these ones. The former case means that the lower bound (relaxed solution) represents an integer solution (like the upper bound). While for the upper bound the optimal value of bilinear terms just can be selected from the extreme of partitions. Besides other difference between the partition schemes is related to their numbers of partitions, where the number of partitions for the lower bound is always less than the number of partitions for the upper bound. Finally, the steps for the spatial branch and bound algorithm proposed are the following:

Step 1: Determine from the data the overall domain of the property operators ($\psi_{\mathrm{p}}^{\min}$, $\psi_{\mathrm{p}}^{\max}$),

$$\psi_{\mathrm{p}}^{\min} \le \psi_{\mathrm{p}} \le \psi_{\mathrm{p}}^{\max}$$

Step 2: Determine the number of partitions for each property operator in the upper and lower bounds of each new subdomain, as it can be seen in Figure 9, which presents two subdomains and five partitions for these ones.

Step 3: Solve the upper and lower bounds for each new domain and select the best solution for the upper and lower bounds in all domains, the lowest for the upper bound and the highest for the lower bound. If the gap is lower than the minimum allowed difference, the upper bound represents the global optimal solution and the search is stopped.

Step 4: If the gap is higher than the minimum allowed difference, then select the values of the property operators and the flow rates at the inlet to the interceptors for the upper bound in each domain and use them for fixing two new domains respect to the property operators: one for the lower limit and the other for the upper limit of each domain using the flow rate interceptors like lower and upper limits for each one (see Figure 5c). Eliminate the section when the relaxed problem has a higher solution than the best integer solution.

$$\psi_{\mathrm{p,t}}^{\min} \le \psi_{\mathrm{p,t,left}} = \psi_{\mathrm{p,t}} = \psi_{\mathrm{p,t,right}} \le \psi_{\mathrm{p,t}}^{\max}$$

Therefore, the new domains are,

$$\psi_{\mathrm{p,t-1}}^{\min} \le \psi_{\mathrm{p,t}} \le \psi_{\mathrm{p,t-1,left}}$$

$$\psi_{\mathrm{p,t-1,right}} \le \psi_{\mathrm{p,t}} \le \psi_{\mathrm{p,t-1}}^{\max}$$

Notice that the subscript for ψ_{p} has changed in this expression (t instead of $t-1$) to show that it corresponds to the following iteration, but the operator is the same only with different limits.

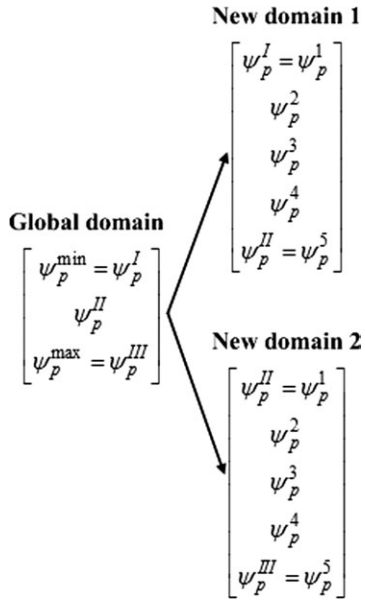

Figure 9. Decomposition of the global domain into subdomains.

Step 5. With the new domains, solve the model for the upper and lower bounds, and select the best solutions for the upper and lower bounds; then, compare them. If the minimum gap is reached, the search is stopped; otherwise, select the values of the upper bound for the property operators and the flow rates at the interceptors and return to Step 3. But now the new two domains must be fixed over the domain that shows the best solution for the upper bound, as can be seen in Figure 8b. For example, if the best solution was obtained from the left domain, the new two domains for the next iteration are,

$$\psi_{\mathrm{p,t-1}}^{\min} \le \psi_{\mathrm{p,t,left}} = \psi_{\mathrm{p,t}} = \psi_{\mathrm{p,t,right}} \le \psi_{\mathrm{p,t-1,left}}$$

$$\psi_{\mathrm{p,t-2}}^{\min} \le \psi_{\mathrm{p,t}} \le \psi_{\mathrm{p,t-1,left}}$$

$$\psi_{\mathrm{p,t-1,right}} \le \psi_{\mathrm{p,t}} \le \psi_{\mathrm{p,t-2,left}}$$

Again, the subscript t for ψ indicates a new iteration.

Step 6. This partitioning scheme is repeated until the minimum allowed gap between the upper and lower bound is reached.

It should be noted that the proposed algorithm incorporates two novel ideas. One of them is the convex relaxations and the other one is the branching variable rules. In this regard, both strategies are strongly related into the iterative method. This is because the results of one are used in the other one; for example, the first step for the convex relaxation is to identify the subdomains, where the results obtained in these areas are used for the branching rules (see Figure 8). In addition, one of the steps in the branching rules is to determine the new zones for the new convex relaxation. Therefore, both the convex relaxations and the branching variable rules are important related steps that impact positively on the proposed algorithm.

Finally, the optimal design of EIP has positive impacts simultaneously on the economy and the environment. This integral approach reduces simultaneously the associated costs (because of the reduction in the fresh water consumption and

Table 2. Data for Example 1

Plants	Sinks	Flow Rate (kg/h)	Composition (ppm) Min	Composition (ppm) Max	Toxicity (%) Min	Toxicity (%) Max	Plants	Sources	Flow Rate (kg/h)	Composition (ppm)	Toxicity (%)
1	1	2900	0	0.013	0	1.500	1	1	2900	0.033	1.800
	2	2450	0	0.011	0	0.750		2	2450	0.022	0.500
	3	8083	0	0.013	0	1.250		3	8083	0.016	2.300
2	4	3900	0	0.011	0	1.750	2	4	3900	0.024	1.500
	5	3279	0	0.100	0	1.150		5	3279	0.220	1.500
	6	3100	0	0.100	0	0.800		6	3100	0.010	0.750
3	7	1800	0	0.010	0	0.950	3	7	1800	0.160	1.400
	8	1750	0	0.040	0	0.750		8	1750	0.100	1.750
	9	2000	0	0.020	0	1.250		9	2000	0.110	1.300
Waste		–	0	0.075	0	0.000	Fresh water		–	0.000	0.000

Property	Interceptors	CU_r (US$)	CUMr (US$/kg)	RR
Composition	1	7500	0.0065	0.02
	2	5000	0.0033	0.15
Toxicity	3	9200	0.0098	0.00

treatment requirements) and the environmental impact (because of the reduction in the overall wastewater discharged to the environment) by sharing treatment units and pipes among participating plants to develop an EIP. To solve this problem as a multiobjective optimization problem, the presented model can consider the simultaneous minimization of the associated costs and the fresh water consumption. This way, the problem can be solved using the constraint method, where the minimization for the associated costs can be considered as an objective and the maximum fresh water consumption can be considered as an additional constraint (or the maximum wastewater discharged to the environment). This problem can be solved for several constraints to determine a Pareto curve.

Results

Two examples are presented to show the application of the proposed algorithm. The data for these examples were proposed in this article from typical values found in the process industry. The values for the parameters H_Y, K_F, α, ρ, and v are 8000 h/year, 0.231/year, 0.70, 1000 kg/m^3, and 1 m/s, respectively. The material for the pipes is carbon steel with the cost parameters of p and CU_p equal to 7200 and 250, respectively. The distance for pipe segments between sources and sinks in the same plant is 50 m and for the rest of pipe segments is 200 m; whereas 5 and 21 partitions were used for solving the relaxed problem and integer problem, respectively. Additional numbers of partitions were tested; however, when the number of partitions increases the demanded computational effort in each iteration increases significantly; on the other hand, for lower number of partitions the reformulations are not good enough. Therefore, for small problems and moderated problems, the number of partitions used in this article are good enough. For bigger problems, the number of partitions can be reduced depending on the capacity of the computer used.

Also, like the units of flow rates required to calculate the capital cost for the pipes from Eq. 3 should be given in ton/h, a conversion factor to transform from kg/h (units used in this work for the flow rates) to ton/h was used. Both the integer and relaxed problems were implemented in the software GAMS (Brooke et al.[48]) and the resulting problems were solved using the solver CPLEX for the MILP problems (upper and lower bounds) in a computer with an Intel® Core^{TM} i7-920 at 2.67 GHz and 9.00-GB RAM.

Example 1: This example consists of three plants with three process sources and three process sinks each one, whose data are shown in Table 2. In addition, two properties are considered (composition and toxicity), and one type of pure fresh water is available with a unit cost of 0.009 US$/kg. For this example, the lower limits for the property operators are 0.01 ppm for the composition and 0.5% for the toxicity, while the upper limits for above operators are 0.22 ppm and 2.3%. Therefore, the domains for each property operator are $0.01 \leq \psi_c \leq 0.22$ and $0.50 \leq \psi_{tox} \leq 2.30$, and a partition at the value of 0.115 for the composition and 1.400 for the toxicity were made. Then, the original domain for the composition is covered by two new subdomains: $0.01 \leq \psi_c \leq 0.115$ and $0.115 \leq \psi_c \leq 0.22$, while the original domain of the toxicity is covered by the next two subdomains: $0.50 \leq \psi_{tox} \leq 1.40$ and $1.40 \leq \psi_{tox} \leq 2.30$; and their possible combinations including the original domain are presented in Figure 10. It should be noted that the number of partitions for the decomposition of the original domains should be determined according to the behavior of the global optimization algorithm; in this regard, if the global optimization algorithm does not evolve, then it is needed to increase the number of partitions to improve its performance (i.e., to contract the space for the relaxed solution and to increase the value for the lower bound). However, one partition over the original domain of the properties is a good number from the experience in this work. In addition, Appendix B shows the

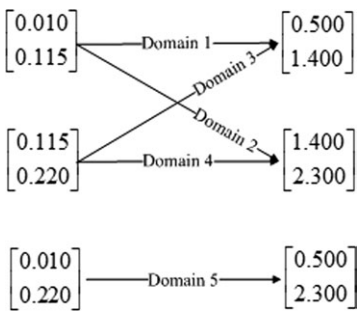

Figure 10. Combinations of the new subdomains for Example 1.

Table 3. Lower and Upper Bound for the Property Operators for Example 1

	Composition		Toxicity	
Domain	Lower Limit	Upper Limit	Lower Limit	Upper Limit
1	0.010	0.115	0.500	1.400
2	0.010	0.115	1.400	2.300
3	0.115	0.220	0.500	1.400
4	0.115	0.220	1.400	2.300
5	0.010	0.220	0.500	2.300

explanation for the advantages of the decomposition of the original domains of the property operators.

Notice that there are five new domains, and the lower and upper limits for the property operators (composition and toxicity) on these ones are shown in Table 3. In addition, the results obtained through the iterations are presented in Table 4 together with the original domain which is called the Domain 5. For the first iteration, the best solution for the integer feasible problem (upper bound) was found in the Domain 5, while for the relaxed problem it was found in the Domain 1. The gap between the upper and lower bounds is 29.54%. But, in the Domain 3, the left relaxed solution is bigger than the best integer solution; therefore, this section is eliminated. Also, for the Domain 4, it is not found any feasible solution for the integer problem and its relaxed solution is bigger than the best integer solution; therefore, this section is eliminated, too. In the second iteration, there are only three domains for searching and, in this iteration, the best solutions for the integer and relaxed problems were obtained in the Domain 5 with a gap of 1.15%. Also, both the left and right lower bounds for the Domain 1 are bigger than the best upper bound; then, this zone is eliminated for the next iterations. For the third iteration only two domains are available, and the upper bounds were improved in a small percent; again the best upper and lower bounds were found in the Domain 5 with a gap of 1.03%. In the fourth iteration, the upper bound was not improved, but the lower bound had a considerable increment to get a gap of 0.70%, so the search is stopped. Finally, the numerical behavior of the proposed procedure is shown in Figure 11, where the optimal solution is obtained at the Iteration 4 that provides a gap between the upper and lower bound by 0.70%. In addition, for determining the effect of the number of partitions of the relaxed problem on the proposed procedure, this problem was solved with 10 and 3 partitions obtaining the results that are shown in Table 5 and its corresponding graphical representation presented in Figure 11. Notice that with 10, 5, and 3 partitions, the overall convergence was reached in the fourth iteration with a gap of 0.46, 0.70, and 0.59%, respectively; however, for 10 and 5 partitions the gap is very small since the Iteration 2.

Figure 12 shows the optimal configuration for this example. Notice that it was required to treat both properties in the shared treatment units, the composition in the Interceptor 1 and the toxicity in the Interceptor 3. Notice also that recycle streams between the interceptors are not needed. In addition, the optimal EIP presents both single plant mass exchange and interplant mass exchange because portions of the Sources 1, 2, and 3 of the Plant 1 are sent to the Plants 2 and 3, Sources 4 and 6 of the Plant 2 are split and sent to the Plants 1 and 3, and portions of Sources 7 and 9 of the Plant 3 are sent to the Plant 2. This shows that the proposed model for the interplant integration reduces significantly the fresh water usage and, at the same time, the waste discharged to the environment, because the reuse of process sources of each plant in different plants avoids to send them to the waste and allows to meet the constraints imposed on the process sinks. The total savings for the fresh water consumption and for the wastewater discharged to the environment are both 84% respect to the case without integration. Finally, Table 8 summarizes the results for this example problem; it is important to remark that savings of 15.15, 6.08, and 68% are obtained with the procedure proposed in this article respect to the solution obtained using the solvers DICOPT, branch-and-reduce optimization navigator (BARON), and standard branch and bound (SBB) for the original nonconvex problem, respectively. Notice in Table 8 that the total time to solve the relaxed and integer problems is 11,820 s, while when the solvers DICOPT and BARON are used, the consumed CPU times are 275 and 90,000 s, respectively.

Table 4. Results for Each Domain in the Iterations for Example 1

		Bounds				Best Solutions of Bounds in Each Domain		% Gap in Each Domain	Overall Best Solutions		% Overall Gap
		Upper		Lower							
Iterations	Domains	Left	Right	Left	Right	Upper	Lower		Upper	Lower	
1	1	2,977,549	2,977,549	1,799,947	1,799,947	2,977,549	1,799,947	39.55	2,554,648	1,799,947	**29.54**
	2	2,685,041	2,685,041	1,582,036	1,582,036	2,648,738	1,582,036	40.27			
	3	3,305,369	3,305,369	2,705,021	2,705,021	3,305,369	2,705,021	18.16			
	4	Eliminated	Eliminated	3,035,180	3,035,180	Eliminated	Eliminated	Eliminated			
	5	2,554,648	2,554,648	789,503	789,503	2,554,648	789,503	69.10			
2	1	2,976,860	2,975,398	2,818,512	2,820,541	2,975,398	2,820,541	5.20	2,486,352	2,457,661	**1.15**
	2	2,606,893	2,685,076	2,286,383	2,354,349	2,606,893	2,354,349	9.69			
	3	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated			
	4	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated			
	5	2,582,269	2,486,352	2,457,661	2,509,239	2,486,348	2,457,661	1.15			
3	1	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	2,486,348	2,460,681	**1.03**
	2	2,605,620	2,608,553	2,409,178	2,597,085	2,605,620	2,597,085	0.33			
	3	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated			
	4	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated			
	5	2,486,352	2,486,348	2,427,971	2,460,681	2,486,348	2,460,681	1.03			
4	1	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	2,486,348	2,469,001	**0.70**
	2	2,605,620	2,608,553	2,540,567	2,597,085	2,605,620	2,597,085	0.33			
	3	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated			
	4	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated			
	5	2,486,352	2,486,348	2,469,001	2,495,015	2,486,348	2,469,001	0.70			

Table 5. Sensitivity Analysis Varying the Number of Partitions in the Relaxed Problem of the Example 1

Number of Partitions	Iterations	Best Solution: Upper Bound	Best Solution: Lower Bound	% Gap
10	1	2,554,648	2,123,470	16.88
	2	2,486,352	2,457,661	1.15
	3	2,486,348	2,474,798	0.46
	4	2,486,348	2,474,798	**0.46**
5	1	2,554,648	1,799,947	29.54
	2	2,486,352	2,457,661	1.15
	3	2,486,348	2,460,681	1.03
	4	2,486,348	2,469,001	**0.70**
3	1	2,554,648	930,496	63.58
	2	2,486,352	2,194,078	11.76
	3	2,486,348	2,427,462	2.37
	4	2,486,348	2,471,643	**0.59**

Therefore, the optimal solution for this problem was found by the proposed strategy in much less CPU time (3.3 h) than the one required by other global optimization solvers (the best solution found with BARON demanded 25 h and the solver SBB demanded 9.3 h).

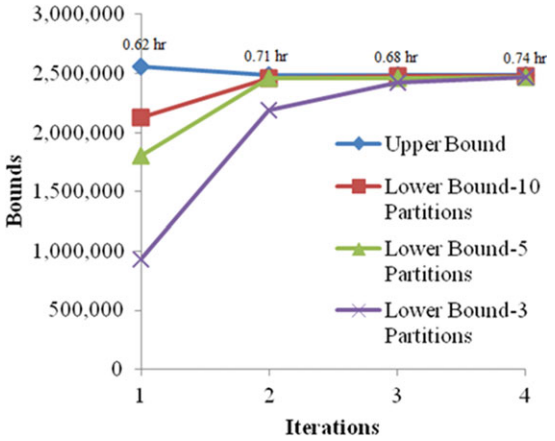

Figure 11. Evolution of the global optimization approach for Example 1 as a function of the number of partitions for the relaxed problem.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com]

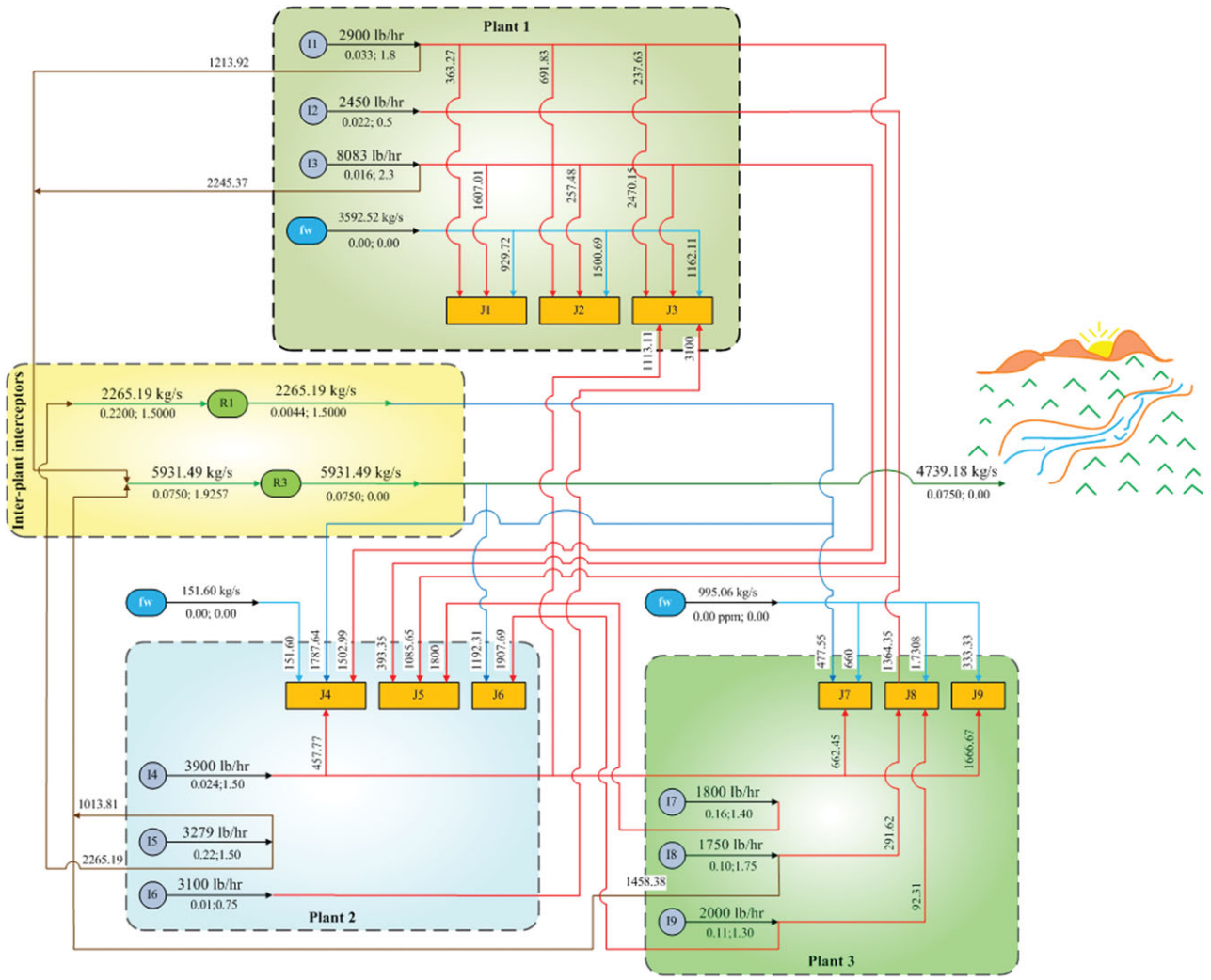

Figure 12. Optimal configuration for Example 1.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com]

Table 6. Data for Example 2

Plants	Sinks	Flow Rate (kg/h)	Toxicity (%) Min	Toxicity (%) Max	pH Min	pH Max	THOD (mg O_2/L) Min	THOD (mg O_2/L) Max	Plants	Sources	Flow Rate (kg/h)	Toxicity (%)	pH	THOD (mg O_2/l)
										1	35,500.0	0.95	6.00	60.00
1	1	35,500	0.00	2.00	5.10	8.50	0.00	75.00		2	10,000.0	1.75	5.40	88.00
	2	10,000	0.00	0.75	5.20	8.50	0.00	75.00		3	50,000.0	0.50	5.50	60.00
	3	50,000	0.00	1.30	5.10	8.50	0.00	100.00	2	4	45,800.0	1.50	5.40	65.00
2	4	45,800	0.00	1.00	5.20	8.00	0.00	75.00		5	50,000.0	0.85	6.00	100.00
	5	50,000	0.00	0.95	5.30	8.00	0.00	75.00		6	40,000.0	1.70	5.80	80.00
	6	40,000	0.00	1.25	5.25	8.50	0.00	65.00	3	7	55,000.0	1.30	6.00	85.00
3	7	55,000	0.00	0.50	5.30	8.00	0.00	100.00		8	11,000.0	0.50	5.90	60.00
	8	11,000	0.00	2.00	5.20	8.50	0.00	65.00		9	85,000.0	1.00	5.50	100.00
	9	85,000	0.00	1.20	5.20	7.00	0.00	65.00	Fresh water 1		–	0.00	7.00	0.00
Waste		–	0.00	0.00	5.80	9.00	0.00	60.00	Fresh water 2		–	0.01	6.80	0.00

Interceptors	CUr (US$)	CUMr (US$/kg)	RR Toxicity	RR pH	RR THOD
1	9200	0.0098	0.00	0.00	0.00
2	8500	0.0075	0.10	0.00	0.00
3	9000	0.0063	0.00	0.01	0.00
4	8000	0.0032	0.00	0.10	0.00
5	8700	0.0065	0.00	0.00	0.20
6	5500	0.0032	0.00	0.00	0.45

Table 7. Results for Each Domain in the Iterations for Example 2

Iterations	Domains	Bounds Upper Left	Bounds Upper Right	Bounds Lower Left	Bounds Lower Right	Best Solutions of Bounds in Each Domain Upper	Best Solutions of Bounds in Each Domain Lower	% Gap in Each Domain	Overall Best Solutions Upper	Overall Best Solutions Lower	% Overall Gap
1	1	5,868,905	5,868,905	3,838,974	3,838,974	5,868,905	3,838,974	34.59	5,003,684	4,719,422	**5.681**
	2	5,033,684	5,033,684	3,849,783	3,849,783	5,033,684	3,849,783	23.52			
	3	5,909,783	5,909,783	5,113,788	5,113,788	5,909,783	5,113,788	13.47			
	4	5,030,477	5,030,477	3,719,836	3,719,836	5,030,477	3,719,836	26.05			
	5	5,964,789	5,964,789	4,719,422	4,719,422	5,964,789	4,719,422	20.88			
	6	6,770,321	6,770,321	4,325,408	4,325,408	6,770,321	4,325,408	36.11			
	7	5,956,786	5,956,786	4,710,761	4,710,761	5,956,786	4,710,761	20.92			
	8	5,265,105	5,265,105	4,229,555	4,229,555	5,265,105	4,229,555	19.67			
	9	5,003,684	5,003,684	2,181,809	2,181,809	5,003,684	2,181,809	56.40			
2	1	5,853,277	5,823,408	5,814,040	5,831,368	5,823,408	5,814,040	0.16	4,903,594	4,897,328	**0.128**
	2	4,942,573	4,959,345	4,000,881	4,051,797	4,942,573	4,051,797	18.02			
	3	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated			
	4	5,030,616	4,903,594	3,729,895	3,899,218	4,903,594	3,899,218	20.48			
	5	6,153,780	5,964,789	5,941,692	4,897,328	5,964,789	5,941,692	0.39			
	6	5,210,869	6,709,677	5,152,412	5,144,055	5,210,869	5,152,412	1.12			
	7	8,405,846	5,947,795	5,936,765	5,916,969	5,947,795	5,936,765	0.19			
	8	5,265,105	5,266,294	5,236,229	5,247,784	5,265,105	5,247,784	0.33			
	9	4,922,037	4,922,037	4,919,150	4,884,500	4,922,037	4,919,150	0.06			
3	1	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	4,903,594	4,903,446	**0.003**
	2	4,942,573	4,959,345	4,903,446	4,939,688	4,942,573	4,939,688	0.06			
	3	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated			
	4	5,030,616	4,903,594	4,871,833	4,903,594	4,903,594	4,903,594	0.00		4,903,594	**0.000**
	5	6,153,780	5,964,789	5,941,692	4,897,328	5,964,789	5,941,692	0.39			
	6	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated			
	7	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated		4,884,500	**0.389**
	8	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated	Eliminated			
	9	4,922,037	4,922,037	4,919,150	4,884,500	4,922,037	4,919,150	0.06			

Example 2: The second example considers three plants with three process sources and three process sinks each one. The properties allowed to be treated are toxicity, pH, and theoretical oxygen demand (THOD). There are two types of fresh water available with unit costs of 0.009 US$/kg and 0.0075 US$/kg for fresh waters 1 and 2, respectively. The data for the process sources, process sinks, fresh waters, and interceptors are shown in Table 6.

The limits for the property operators for toxicity, pH, and THOD can be determined from the data as 0.50, 251, 188.6, and 60 for the lower bounds, and 1.75, 1,000,000, and 100 for the upper bounds, respectively. Each domain was divided into two subdomains and there are eight combinations between them; therefore, there are eight new domains together with the original domain that is called the Domain 9. The solution for each iteration is shown in Table 7. In the first iteration, the best upper bound was found in the Domain 9, whereas the best lower bound was found in the Domain 5; the corresponding gap was 5.68%. Also, in this iteration, the lower bound in the Domain 3 was bigger than the best lower bound and, therefore, this domain was eliminated for the next iteration. In the second iteration, the best upper and

Table 8. Results for Examples

	Integer Problem	Relaxed Problem	DICOPT	BARON	SBB
Example 1					
Total annual cost (US$/year)	2,486,348	2,469,001	2,930,415	2,647,517	7,838,609
Fresh water cost (US$/year)	341,221	318,214	203,394	473,690	1,696,464
Regeneration cost (US$/year)	1,899,687	1,905,347	2,455,593	1,931,274	6,009,315
Pipeline cost (US$/year)	245,441	245,441	271,428	242,553	132,830
Constraints	8424	4416	408	408	408
Binary variables	514	242	166	166	166
Continuous variables	6031	1727	391	391	391
CPU (s)	9900	1920	275	90,000	33,372
Example 2					
Total annual cost (US$/year)	4,903,594	4,903,594	7,494,930	No feasible solution	21,416,890
Fresh water cost (US$/year)	0	0	0		0
Regeneration cost (US$/year)	4,637,908	4,637,908	7,243,686		21,177,180
Pipeline cost (US$/year)	265,686	265,686	251,245		239,711
Constraints	25,782	13,278	610		610
Binary variables	1,126	412	223		223
Continuous variables	18,148	4,708	592		592
CPU (s)	18,900	755	540	86,400	85,932

lower bounds were determined in the Domains 4 right and 5 right with a gap of 0.13%, which is inside of the minimum allowed gap. In addition, notice that in this iteration the overall minimum gap for the Domains 1, 5–9 was reached, and the lower bound for the Domains 1, 6–8 on the left and right hand sides are bigger than the best lower bound and, therefore, for the next iteration, these sections were eliminated. Although the minimum gap was reached, a third iteration was done for considering the possibility to improve the upper bound, but this did not occur. However, the lower bound for the Domains 2 and 4 were improved and they can be valid lower bounds because the Domain 2 left represents a gap of 0.003% compared with the best lower bound (4 right) and the lower bound in the Domain 4 left represents a gap of almost zero. Finally, in this example was found the minimum gap for every new domain except for the domain three because it was eliminated in the first iteration, and the overall best lower bound can be represented by three Domains (5 right, 2 left, and 4 right). The behavior for the upper and lower bounds are showed in Figure 13.

The optimal configuration for this example is presented in Figure 14, which shows single plant mass exchange and interplant mass exchange from Plant 1 to Plants 2 and 3, from Plant 2 to Plants 1 and 3, and from Plant 3 to Plants 1 and 2. Only one interceptor was selected to treat the THOD and there is a recycled stream in the regeneration zone from the false interceptor to Interceptor 6. There is no waste discharged to the environment. Notice that without considering the interplant integration, the reused portions of process sources of each plant in other plants would have been sent to the waste and, therefore, it would have required fresh water to satisfy the process sinks constraints. Notice that the solution of Figure 14 does not require fresh water; therefore, the total savings in fresh water consumed and wastewater discharged to the environment are 100%. The costs, problem size, and CPU time for the integer problem, the relaxed problem, and the original problem with solvers DICOPT, BARON, and SBB are shown in Table 8. The costs of the optimal configurations obtained with the solvers DICOPT and SBB are 34.57 and 77% more expensive than the solution obtained in this article, respectively. On the other hand, BARON was not able to provide a feasible solution even consuming a huge computational time (around 24 h) because of the great number of bilinear terms and the difficulty to find good initial values and bounds for the major variables of the proposed model. In addition, the solution obtained with the solver SBB consumes 23.9 h. Therefore, we demonstrate that the computational time required by the proposed optimization algorithm to get the solution (5.5 h) is reasonable. Therefore, this approach improves the results obtained with traditional techniques. Furthermore, the gap in the last iteration with 10 and 3 partitions for this example are 0.003 and 0.118%, respectively.

The total computation time shown in Table 8 is disaggregated in the CPU time consumed in each iteration in Figures 11 and 13 for the Examples 1 and 2, respectively.

Conclusions

A model for the property-based interplant water integration was proposed. The model is based on a general superstructure that allows simultaneously the single plant and interplant water integration, in addition to a shared treatment system to improve the properties of the streams considering the possible integrations previously avoided because the numerical complications that they involve. The objective

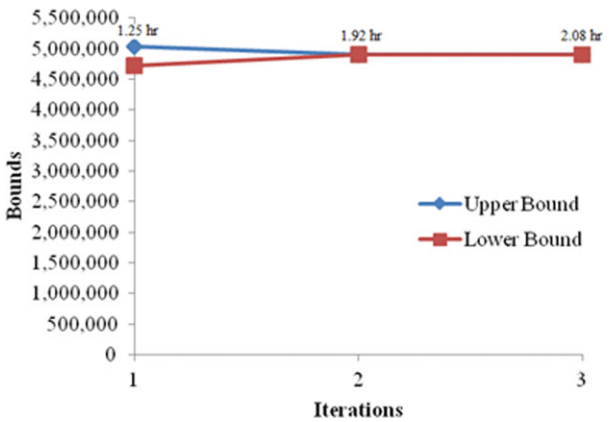

Figure 13. Evolution of the global optimization approach for Example 2.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com]

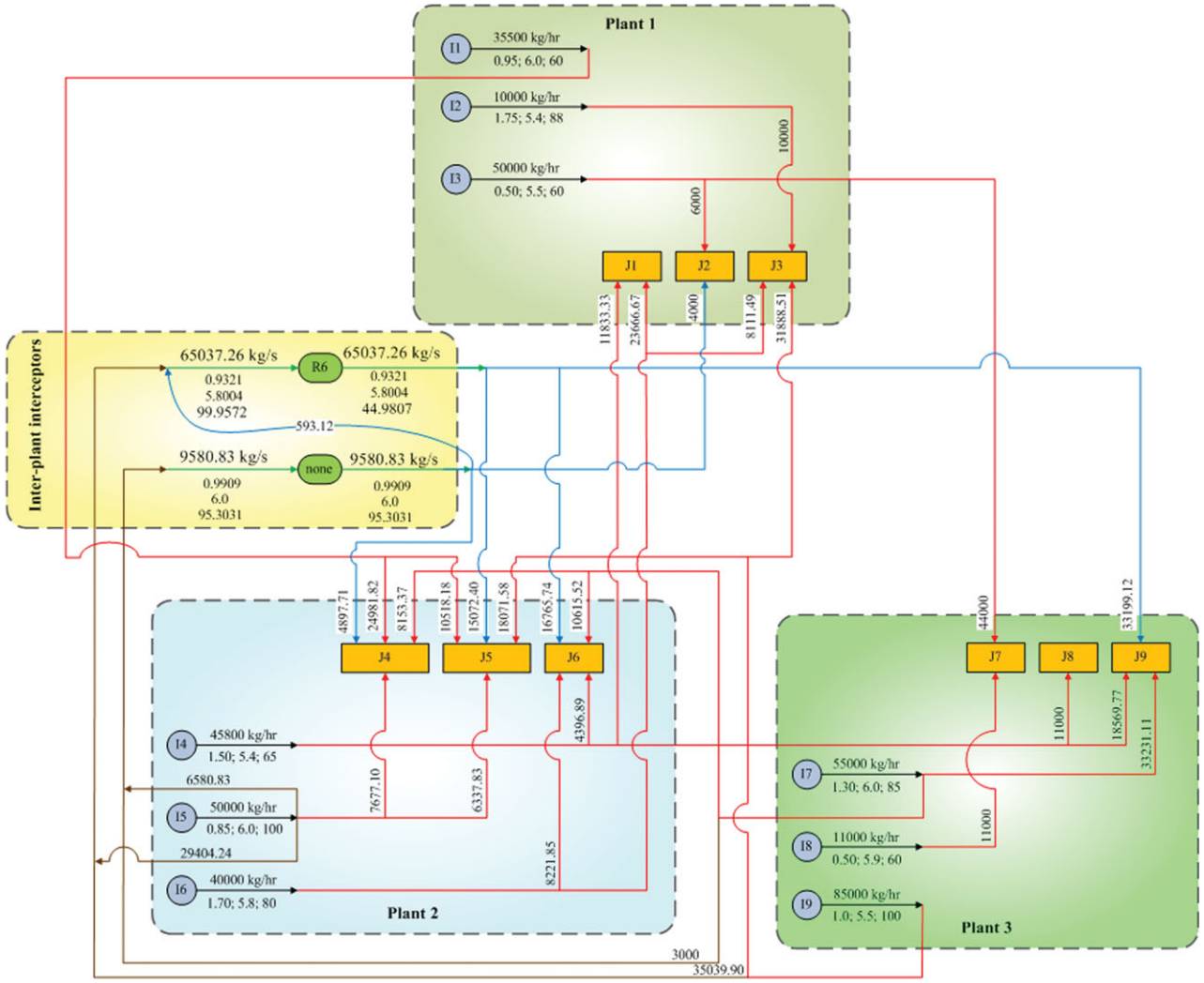

Figure 14. Optimal configuration for Example 2.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com]

function consists in minimizing the total annual cost constituted by the fresh water cost, regeneration cost, and piping cost.

Unlike most of the previously reported methods for the synthesis of interplant water networks, the proposed model is based on the properties of the streams, situation that is very useful for cases constituted by several pollutants.

Because the model for the interplant water integration consists of a lot of bilinear terms, this article proposed a new spatial branch and bound procedure based on two new reformulations for the upper and lower bounds. This algorithm consists in solving iteratively the upper bound (integer feasible solution) and a lower bound (relaxed solution) until the gap between them is lower than a given tolerance considering the partition over the original domain into several subintervals. This article also proposes new branching rules for the addressed problem to reach the global optimal solution in few iterations.

The application of the proposed algorithm to two examples shows that the proposed interplant water integration superstructure is able to yield significant savings in the fresh water consumption and, at the same time, to reduce the overall waste discharged to the environment yielding solutions economical and environmentally attractive. The model based on properties avoids handling a lot of compositions for the different components presented in the waste streams to determine the properties that are restricted by the process sinks and the environment. Finally, the proposed global optimization approach allows finding the global optimal solution in a few iterations and in a reasonable CPU time, which is very useful for interplant water integration problems in which a large number of bilinear terms are presented.

Acknowledgments

The authors acknowledge SEP-CONACYT Grant 130207 for financial support, and J.M. Ponce-Ortega and E. Rubio-Castro also acknowledge the financial support from CONACYT for the scholarships provided.

Notation

Parameters

A_z = slope of lineal segments
C_z = intersection of lineal segments
CUM_r = unit cost for mass removed in each interceptor, US$/kg
CU_p = pipe unit cost, US$
CU_r = investment cost coefficient for interceptors, US$
CU_w = fresh water unit cost, US$/kg
$D_{k,p}$ = domain of the property operators
$D^l_{i,j}$ = distance between source i and sink j, m

$D^2_{i,r}$ = distance between source i and interceptor r, m
$D^3_{r,j}$ = distance between interceptor r and process sink j, m
D^4_r = distance between interceptor r and environmental discharge, m
D^5_i = distance between source i and environmental discharge, m
$F^{2,max}_k$ = upper bound for F^2_k, kg/h
$F^{2,min}_k$ = lower bound for F^2_k, kg/h
FS_i = flow rate of process source i, kg/h
FU_j = flow rate of process source j, kg/h
H_Y = plant operating hours per year, h/year
K_F = factor used to annualize the capital costs, 1/year
$M_{B_r}{}^{3max}$ = upper limit for $B_{r,z}{}^3$, kg/h
$M_{B_{k,p}}{}^{1max}$ = upper limit for $B^1_{k,p}$
M^{max}_{Fk} = upper limit for F^1_k, kg/h
M^{max}_{fier} = upper limit for the pipe segment from interceptor r to the waste discharged to the environment, kg/h
$M^{max}_{fisr,j}$ = upper limit for the pipe segment from interceptor r to process source j, kg/h
M^{max}_{fsei} = upper limit for the pipe segment form process source i to the waste discharged to the environment, kg/h
$M^{max}_{fsii,r}$ = upper limit for the pipe segment from process source i to interceptor r, kg/h
$M^{max}_{fssi,j}$ = upper limit for the pipe segment from process source i process sink j, kg/h
M^{min}_{fier} = lower limit for the pipe segment from interceptor r to the waste discharged to the environment, kg/h
$M^{min}_{fisr,j}$ = lower limit for the pipe segment from interceptor r to process source j, kg/h
M^{min}_{fsei} = lower limit for the pipe segment from process source i to the waste discharged to the environment, kg/h
$M^{min}_{fsii,r}$ = lower limit for the pipe segment from process source i to interceptor r, kg/h
$M^{min}_{fssi,j}$ = lower limit for the pipe segment from process source i to process sink j, kg/h
$M^{max}_{FRr,z}$ = upper limit for $\delta^3_{r,z}$, kg/h
$M^{min}_{FRr,z}$ = lower limit for $\delta^3_{r,z}$, kg/h
p = parameter for cross-plant pipeline capital cost
$RR_{r,\psi_p(Pp)}$ = efficiency factor of interceptor r for property operator p, dimensionless
v = velocity, m/s
α = cost function exponent, dimensionless
ρ = water density, kg/m^3
ψ^{min}_P = lower limit for property operator p
ψ^{max}_P = lower limit for property operator p
$\psi^{2,max}_{k,p}$ = upper bound for $\psi^2_{k,p}$
$\psi^{2,min}_{k,p}$ = lower bound for $\psi^2_{k,p}$
$\psi_p(Pe^{max}_p)$ = upper limit for the property operator p in the waste discharged to the environment
$\psi_p(Pe^{min}_p)$ = lower limit for the property operator p in the waste discharged to the environment
$\psi_p(Ps_{p,i})$ = property operator p in the process source i
$\psi_p(Pu^{max}_{p,j})$ = upper limit for the property operator p in the process sink j
$\psi_p(Pu^{min}_{p,j})$ = lower limit for the property operator p in the process sink j
$\psi_p(Pw_{p,w})$ = property operator p in the fresh water type w
$\gamma^1_{k,p,q1}$ = discrete value of $\psi^1_{k,p}$
$\gamma^2_{k,p,q2}$ = discrete value of $\psi^2_{k,p}$

Variables

$A_{r,z}$ = disaggregated slope for A_r
$B^1_{k,p}$ = variable that substitutes any bilinear term in the upper bound
$B^2_{k,p}$ = variable that substitutes any bilinear term in the lower bound
B^3_r = variable that substitutes the exponential term FR^{α}_r, kg/h
$C_{r,z}$ = disaggregated intersection for C_r
F^1_k = any flow rate in the upper bound, kg/h
F^2_k = any flow rate in the lower bound, kg/h
FE = flow rate in the waste discharged to the environment, kg/h
fie_r = flow rate from interceptor r to waste discharged to the environment, kg/h
$fii_{r_1,r}$ = flow rate from interceptor r_1 to interceptor r, kg/h
$fis_{r,j}$ = flow rate from interceptor r to process sink j, kg/h
FR_r = flow rate in the interceptor r, kg/h
fse_i = flow rate from process source i to waste discharged to environment, kg/h
$fsi_{i,r}$ = flow rate from process source i to interceptor r, kg/h
$fss_{i,j}$ = flow rate from process source i to process sink j, kg/h
$fws_{w,j}$ = flow rate of the fresh water w in the process sink j, kg/h
PC = cross-plant pipeline capital cost, US$/year
RC = regeneration cost, US$/year
TAC = total annual cost, US$/year
WC = fresh water cost, US$/year
$\beta^1_{k,p,q1}$ = disaggregated bilinear term for $B^1_{k,p}$
$\beta^2_{k,p,q2}$ = disaggregated bilinear term for $B^2_{k,p}$
$\beta^3_{r,z}$ = disaggregated bilinear term for B^3_r
ψ_p = property operator p
$\psi^1_{k,p}$ = property operator in the upper bound
$\psi^2_{k,p}$ = property operator in the lower bound
$\psi_p(Pi^{in}_{p,r})$ = property operator p in the inlet of the interceptor r
$\psi_p(Pi^{out}_{p,r})$ = property operator p in the outlet of the interceptor r
$\delta^1_{k,p,q1}$ = disaggregated flow rate for F^1_k, kg/h
$\delta^2_{k,p,q2}$ = disaggregated flow rate for F^2_k, kg/h
$\delta^3_{r,z}$ = disaggregated flow rate for FR_r, kg/h
$\pi^2_{k,p,q2}$ = disaggregated property operator for $\psi^2_{k,p}$

Binary variables

$x^1_{i,j}$ = binary variable to select the pipe segment from process source i to process source j, 0 or 1
$x^2_{i,r}$ = binary variable to select the pipe segment from process source i to interceptor r, 0 or 1
$x^3_{r,j}$ = binary variable to select the pipe segment from interceptor r to process source j, 0 or 1
x^4_r = binary variable to select the pipe segment from interceptor r to the waste discharged to the environment, 0 or 1
x^5_i = binary variable to select the pipe segment from process source i to the waste discharged to the environment, 0 or 1
$y^1_{k,p,q1}$ = binary variable to select the optimal value of the bilinear term $B^1_{k,p}$, 0 or 1
$y_{k,p,q2}2$ = binary variable to select the optimal value of the bilinear term $B^2_{k,p}$, 0 or 1
$y^3_{r,z}$ = binary variable used to select the optimal value of $B^3_{r,z}$

Subscripts

i = process sources
j = process sinks
k = any flow rate
p = properties
q^1 = partitions in the upper bound
q^2 = partitions in the lower bound
r = property interceptors
t = iterations
w = types of fresh water
z = lineal segments

Superscripts

min = lower limit
max = upper limit

Sets

I = {i=1, 2, …, $N_{sources}$| I is a set of process sources}
J = {j=1, 2, …, N_{sinks}| J is a set of process sinks}
P = {p=1, 2, …, $N_{property\ operators}$| P is a set of property operators}
Q^1 = {q^1=1, 2, …, $N_{partitions}$| Q^1 is a set of partitions of the property operators in the upper bound}
Q^2 = {q^2=1, 2, …,$N_{partitions}$| Q^2 is a set of partitions of the property operators in the lower bound}
R = {r=1, 2, …, $N_{interceptors}$| R is a set of property interceptors}
W = {w=1, 2, …, $N_{type\ of\ freshwater}$| W is a set of fresh water types}
Z = {z=1, 2, 3| Z is a set of lineal segments for calculating the variable capital cost of interceptors}

Scalars

n^1 = number for splitting the property operators for the upper bound
n^2 = number for splitting the property operators for the lower bound

Literature Cited

1. Bagajewicz M. A review of recent design procedures for water networks in refineries and process plants. *Comput Chem Eng.* 2000;24: 2093–2113.

2. Dunn RF, El-Halwagi MM. Process integration technology review: background and applications in the chemical process industry. *Chem Technol Biotechnol*. 2003;78:1011–1021.
3. Foo DCY. State-of-the-art review of pinch analysis techniques for water network synthesis. *Ind Eng Chem Res*. 2009;48:5125–5159.
4. Olesen SG, Polley GT. Dealing with plant geography and piping constraints in water network design. *Trans Inst Chem Eng*. 1996;74:273–276.
5. Spriggs D, Lowe E, Watz J, El-Halwagi MM, Lovelady EM. Design and development of eco-industrial parks. In: AIChE Spring Meeting, New Orleans, LA, 2004.
6. Chew IML, Foo DCY, Ng DKS. *Targeting for plant-wide water integration*. In: *Joint Symposium for Chemical and Metallurgical Engineering*, Pretoria, South Africa. 2007.
7. Foo DCY. Flowrate targeting for threshold problems and plant-wide integration for water network synthesis. *J Environ Manage*. 2008;88:253–274.
8. Bandyopadhyay S, Sahu GC, Foo DCY, Tan RR. Segregated targeting for multiple resource networks using decomposition algorithm. *AIChE J*. 2010;56:1235–1248.
9. Chew IML, Foo DCY, Ng DKS, Tan RR. Flowrate targeting for interplant resource conservation network. Part 1: unassisted integration scheme. *Ind Eng Chem Res*. 2010;49:6439–6455.
10. Chew IML, Foo DCY, Tan RR. Flowrate targeting for interplant resource conservation network. Part 2: unassisted integration scheme. *Ind Eng Chem Res*. 2010;49:6456–6468.
11. Lovelady EM, El-Halwagi MM, Krishnagopalan GA. An integrated approach to the optimization of water usage and discharge in pulp and paper plants. *Int J Environ Pollut*. 2007;29:274–307.
12. Liao ZW, Wu JT, Jiang BB, Wang JD, Yang YR. Design methodology for flexible multiple plant water networks. *Ind Eng Chem Res*. 2007;46:4954–4963.
13. Chew IML, Tan R., Ng DKS, Foo DCY, Majozi T, Gouws J. Synthesis of direct and indirect interplant water network. *Ind Eng Chem Res*. 2008;47:9485–9496.
14. Chew IML, Foo DCY. Automated targeting for inter-plant water integration. *Chem Eng J*. 2009;153:23–36.
15. Lovelady EM, El-Halwagi MM. Design and integration of eco-industrial parks for managing water resources. *Environ Prog Sustain Energy*. 2009;28:265–272.
16. Lim SR, Park JM. Interfactory and intrafactory water network system to remodel a conventional industrial park to a green eco-industrial park. *Ind Eng Chem Res*. 2010;49:1351–1358.
17. Chen CL, Hung SW, Lee JY. Design of inter-plant water network with central and decentralized water mains. *Comput Chem Eng*. 2010;34:1522–1531.
18. Aviso KB, Tan RR, Culaba AB. Designing eco-industrial water exchange networks using fuzzy mathematical programming. *Clean Technol Environ Policy* 2010;12:353–363.
19. Aviso KB, Tan RR., Culaba AB, Cruz JB. Bi-level fuzzy optimization approach for water exchange in eco-industrial parks. *Process Saf Environ Prot*. 2010;88:31–40.
20. Rubio-Castro E, Ponce-Ortega JM, Nápoles-Rivera F, El-Halwagi MM, Serna-González M, Jiménez-Gutiérrez A. Water integration of eco-industrial parks using a global optimization approach. *Ind Eng Chem Res*. 2010;49:9945–9960.
21. Rubio-Castro E, Ponce-Ortega JM, Serna-González M, Jiménez-Gutiérrez A, El-Halwagi MM. A global optimal formulation for the water integration in eco-industrial parks considering multiple pollutants. *Comput Chem Eng*. 2011;35:1558–1574.
22. Tan RR, Aviso KB, Cruz JB, Culaba AB. A note on an extended fuzzy bi-level optimization approach for water exchange in eco-industrial parks with hub topology. *Process Saf Environ Prot*. 2011;89:106–111.
23. Taskhiri MS, Tan RR, Chiu ASF. Emergy-based fuzzy optimization approach for water reuse in an eco industrial park. *Resour Conserv Recy*. 2011;55:730–737.
24. Lovelady EM, El-Halwagi MM, Chew IMI, Ng DKS, Foo DCY, Tan RR. *A property-integration approach to the design and integration of eco-industrial parks*. In: El-Halwagi MM, Linninger AA, editors, *Design for Energy and the Environment: Proceedings of the 7th International Conference on the Foundations of Computer-Aided Process Design (FOCAPD)*, Breckenridge, CO, CRC Press, Taylor & Francis, 2009:559–568.
25. Ng DKS, Foo DCY, Rabie A, El-Halwagi MM. Simultaneous synthesis of property-based water reuse/recycle and interception networks for batch processes. *AIChE J*. 2008;54:2634–2632.
26. Ponce-Ortega JM, Hortua AC, El-Halwagi MM, Jiménez-Gutiérrez A. A property-based optimization of direct recycle networks and wastewater treatment processes. *AIChE J*. 2009;55:2329–2344.
27. Ponce-Ortega JM, El-Halwagi MM, Jiménez-Gutiérrez A. Global optimization of property-based recycle and reuse networks including environmental constraints. *Comput Chem Eng*. 2010;34:318–330.
28. Nápoles-Rivera F, Ponce-Ortega JM, El-Halwagi MM, Jiménez-Gutiérrez A. Global optimization of mass and property integration networks with in-plant property interceptors. *Chem Eng Sci*. 2010;65:4363–4377.
29. Shelley MD, El-Halwagi MM. Component-less design of recovery and allocation systems: functionality-based clustering approach. *Comput Chem Eng*. 2000;24:2081–2091.
30. Viswanathan J, Grossmann IE. A combined penalty function and outer approximation method for MINLP optimization. *Comput Chem Eng*. 1990;14:769–782.
31. Quesada I, Grossmann IE. Global optimization of bilinear process networks with multicomponent flows. *Comput Chem Eng*. 1995;19:1219–1242.
32. Pham V, Laird C, El-Halwagi MM. Convex hull discretization approach to the global optimization of pooling problems. *Ind Eng Chem Res*. 2009;48:1973–1979.
33. El-Halwagi MM, Gabriel F, Harell D. Rigorous graphical targeting for resource conservation via material recycle/reuse networks. *Ind Eng Chem Res*. 2003;42:4319–4328.
34. Kim JK, Smith R. Automated design of discontinuous water systems. *Trans Inst Chem Eng*. 2004;82:238–248.
35. Nemhauser G L, Wolsey LA. *Integer and combinatorial optimization*. New York: John Wiley & Sons, Inc., 1988.
36. Teles JP, Castro PM, Matos HA. Global optimization of water networks design using multiparametric disaggregation. *Comput Chem Eng*. 2012;40:132–147.
37. Wicaksono DN, Karimi IA. Piecewise MILP under- and overestimators for global optimization of bilinear programs. *AIChE J*. 2008;54:991–1008.
38. Visweswaran V, Floudas CA. *New formulations and branching strategies for the GOP algorithm*. In Grossmann IE, editor. *Global Optimization in Engineering Design*, Kluwer Book Series in Nonconvex Optimization and its Applications, *Chapter 3*, New York, Springer, 1995.
39. Al-Khayyal FA, Falk JE. Jointly constrained biconvex programming. *Math Oper Res*. 1983;8:273–286.
40. Maranas CD, Floudas CA. A deterministic global optimization approach for molecular structure determination. *J Chem Phys*. 1994;100:1247–1261.
41. Maranas CD, Floudas CA. Global minimum potential energy conformations of small molecules. *J Global Optim*. 1994;4:135–170.
42. Androulakis IP, Maranas CD, Floudas CA. αBB: a global optimization method for general constrained nonconvex problems. *J Global Optim*. 1995;7:337–363.
43. Liu WB, Floudas CA. A remark on the GOP algorithm for global optimization. *J Global Optim*. 1993;3:519–521.
44. Adjiman CS, Androulakis IP, Maranas CD, Floudas CA. A global optimization method, αBB, for process design. *Comput Chem Eng*. 1996;20:S419–S424. In: European Symposium on Computer Aided Process Engineering-6, Rhodes, Greece.
45. Karuppiah R, Grossmann IE. Global optimization for the synthesis of integrated water systems in chemical processes. *Comput Chem Eng*. 2006;30:650–673.
46. Raman R, Grossmann IE. Modeling and computational techniques for logic based integer programming. *Comput Chem Eng*. 1994; 18:563–578.
47. Sherali HD, Alameddine A. A new reformulation linearization technique for bilinear programming problems. *J Global Optim*. 1992; 2:379–410.
48. Brooke A, Kendrick D, Meeraus A. *GAMS User's Guide*. USA: Scientific Press, 2010.
49. El-Halwagi MM, Hamad AA, Garrison GW. Synthesis of waste interception and allocation networks. *AIChE J*. 1996;42:3087–3101.

Appendix A: Reuse Water Scheme

The mass water integration is used to reduce the consumption of fresh water, the environment discharge, and the process costs. In this regard, the wastewater reuse allows obtaining this goal. Therefore, water used in the process

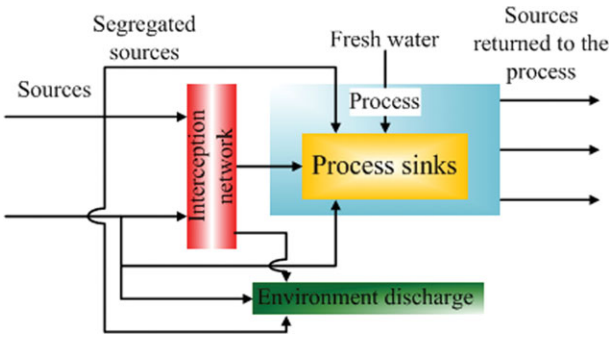

Figure A1. Water reuse scheme.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com]

sinks could be returned to the process (El-Halwagi et al.[49]). From this scheme (see Figure A1), the process sinks are generators of process sources, which can be segregated and sent to the process sinks. This strategy is called direct reuse. On the other hand, the process sources can be segregated and sent to an interception network before the process sinks. In the interceptors, the properties and/or compositions of process sources to reuse them in the process sinks are achieved. This strategy is called indirect reuse. Finally, a little portion of the process sources is sent to the environment discharge where it is mixed with streams coming from the interception network to satisfy the environmental constraints. Notice that without the wastewater reuse just fresh water must be fed to the process sinks.

The process sinks can be heaters, reactors, compressors, and other process units. After processing the streams on these units, the mass of the components could increase or decrease (El-Halwagi et al.[49]). Then, the next property mixing rules express the changes of the properties,

$$\text{Inlet flowrate}^{\text{sink}} \bullet \text{Inlet property}^{\text{sink}} + \Delta^{\text{sink}}_{\text{properties}} = \text{Outlet flowrate}^{\text{sink}} \bullet \text{Outlet property}^{\text{sink}} \quad \text{(A1)}$$

In the above expression, $\Delta^{\text{sink}}_{\text{properties}}$ is the relationship to account for the total change of properties in the process sinks. This relationship could be positive (generation by chemical reaction) or negative (depletion). In the first case, the outlet property is bigger than the inlet property. In the opposite case, the outlet property is lower than the inlet property. In addition, in practice, not always the number of process sinks is equal to the number of process sources recycled to the process for their reuse, because in some cases the outlet flow rates from a specific process sink are sent to different process sinks according to the flow sheet. Therefore, there are two options to characterize the process sources, in terms of flow rate and properties.

Appendix B: Advantages of the Decomposition for the Original Intervals of the Property Operators

The decomposition of the original property operator intervals generates a greater number of optimization problems that must be solved during the iterations of the proposed algorithm. Therefore, to justify such decomposition, we give the following advantages of this decomposition strategy:

• Respect to the integer problem (upper bound), the approximation to the global optimal solution is better solving multiple smaller problems than the original problem. For

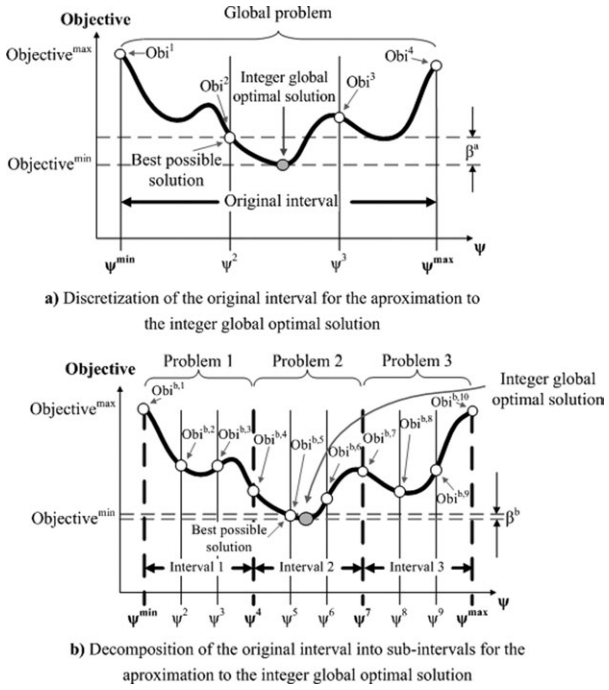

Figure B1. Discretization of the original interval and subintervals of the property operators.

example, if a nonconvex objective function is plotted as a function of one discretized property operator, as it is shown in Figure B1a, three partitions are used to discretize the original interval of the property operator; therefore, there are four possible values for the objective function ($Obj^{a,1}$, $Obj^{a,2}$, $Obj^{a,3}$, and $Obj^{a,4}$) during the optimization process. Notice that the best possible solution for the objective function ($Obj^{a,2}$) corresponds to the second discrete value of the

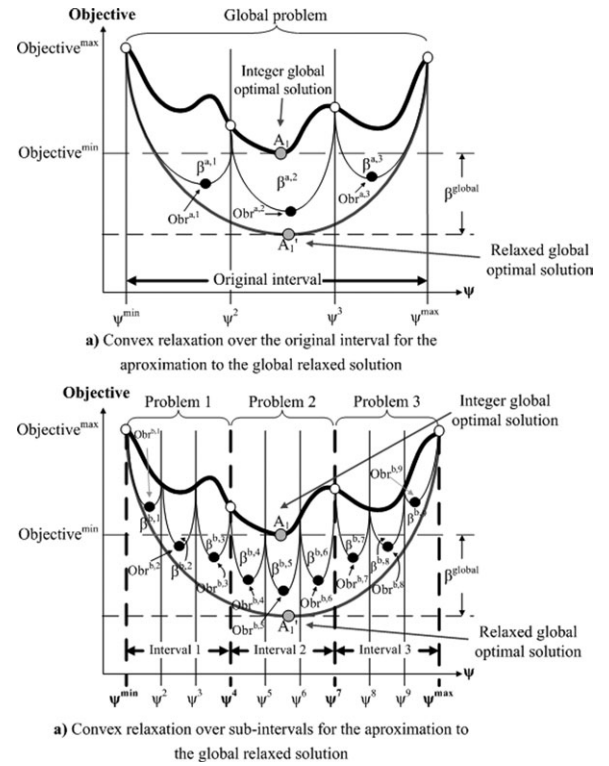

Figure B2. Convex relaxation over the original interval and subintervals of the property operators.

property operator (ψ^2), and this solution has a gap respect to the global optimal solution (β^a). In the other case, when the original interval is divided into three subintervals, three new problems are generated, as can be seen in Figure B1b. In this regard, if also three partitions are used to split each new interval, there are ten possible values for the objective function ($Obj^{b,1}$, $Obj^{b,2}$, $Obj^{b,3}$, $Obj^{b,4}$, $Obj^{b,5}$, $Obj^{b,6}$, $Obj^{b,7}$, $Obj^{b,8}$, $Obj^{b,9}$, and $Obj^{b,10}$) during the optimization process, from which a better approximation to the global solution can be found. And the best possible solution ($Obj^{b,5}$) is found in the fifth discrete value of the property operator (ψ^5), which corresponds to the second discrete value of the property operator in the second new problem. It should be noted that the gap between this solution and the global optimal solution (β^b) is lower than the gap obtained from the scheme shown in Figure B1a (β^a). In addition, for getting the solution $Obj^{b,5}$ without the decomposition of the original interval for the property operator, it must have a number of partitions of 9, which increases the problem size almost three times respect to each new problem shown in Figure B1b. Therefore, it is more difficult to find good solutions without decomposition; this approach also demands more computational efforts. In addition, owing to the great number of variables, the initialization task is very complicated which reduces the possibility to find feasible solutions. If a big number of partitions is considered (in this case 21 partitions were considered), it generates a huge number of problems that are almost impossible to solve in a reasonable CPU time.

- Respect to the relaxed problem (lower bound), Figure B2a presents the possible relaxed solutions of each partition for the original interval of the property operator. As can be seen, there is a gap between such solutions and the integer global solution ($\beta^{a,1}$, $\beta^{a,2}$, and $\beta^{a,3}$). Then, the best relaxed solution is found in the third section, whose gap is represented by $\beta^{a,3}$. In addition, Figure B2b shows the relaxed solutions when the original interval of the property operator is decomposed into three new subintervals. Here, there are nine possible solutions for the relaxed problem ($Obr^{b,1}$, $Obr^{b,2}$, $Obr^{b,3}$, $Obj^{b,4}$, $Obr^{b,5}$, $Obj^{b,6}$, $Obr^{b,7}$, $Obr^{b,8}$, and $Obr^{b,9}$) than can be used as a feasible lower bound during the spatial branch and bound algorithm. It should be noted that the decomposition of the original interval allows improving the gap obtained by partitioning the original interval of the property operator. This is because of a tighter contraction of the feasible zone to find the solution for the relaxed problem. For example, in Figure B2b, the best relaxed solution can be represented by the second ($Obr^{b,2}$) and eight ($Obr^{b,8}$) sections; and these ones have a better gap ($\beta^{b,2}$ and $\beta^{b,8}$) than $\beta^{a,3}$.
- Furthermore, this tighter contraction allows identifying infeasible sections both for the upper bound and the lower bound. In this regard, notice in Figure B2b that the relaxed solutions in the sections one ($Obr^{b,1}$) and nine ($Obr^{b,9}$) are bigger than the best integer solution. Therefore, these sections are eliminated by the proposed algorithm.

Manuscript received Jan. 31, 2012, and revision received May 29, 2012.