Available online at www.sciencedirect.com
SCIENCE DIRECT
Computers and Chemical Engineering 29 (2005) 2198–2214
Computers & Chemical Engineering
www.elsevier.com/locate/compchemeng

Process flowsheet superstructures: Structural multiplicity and redundancy Part II: Ideal and binarily minimal MINLP representations

Tivadar Farkas, Endre Rev*, Zoltan Lelkes

Budapest University of Technology and Economics, H-1111 Budapest, Muegyetem rkp. 3, Budapest, Hungary

Received 9 September 2004; received in revised form 11 July 2005; accepted 25 July 2005

Abstract

The essential problems, namely representativeness and uniqueness, in defining mixed integer non-linear programming (MINLP) representation (MR) is solved in Part I by first defining a basic MR (BMR) that: (1) can be automatically constructed from an easier formable standard GDP representation and (2) serves as a reference representation. Binary and continuous multiplicity of MR are also defined in Part I, and relation is given there between structural redundancy and binary multiplicity.

Based on this results, ideal and binarily minimal MR-s are defined, and the different MR-s are compared from numerical point of view in the present (and final) part. Ideal MR represents all the considered structures and not any other structure. Supposing the process graphs are distincted using binary variables, binarily minimal MR uses the minimal number of them.

Solvability of the different MR-s, including some combined versions, are tested on a middle scale and an industrial scale process synthesis problems. Total solution time, solution time for subproblems, number of iterations, non-ideality and scale of the solvable problems are compared. Idealization of the representation and decreasing the number of binary variables, as suggested in the article, both enhance the solvability and decrease the solution time in a great extent.

Keywords: Multiplicity; Ideality; Redundancy; MINLP; Representation

1. Introduction

Chemical process synthesis consists of three main steps (Grossmann, 1996). First, the considered process structures including all the acceptable unit operations and connections are to be invented and delimited from the potentially infinite number of process structures appropriate to reach the chemical target. This is usually done by constructing a superstructure that somehow includes all the considered structures as its substructures. In the second step, variables representing the possible states of the processes are assigned and an algebraic system, representing a process model, and consisting of equality and inequality constraints acting on the process variables together with some objective function to be optimized is constructed. Finally, in the third step, the optimization task is accomplished, and the results are analyzed.

The graphs that usually are used for representing superstructures may be redundant since their subgraphs may have multiplicity, and therefore, their MINLP model may have different solutions representing the same process. Structural multiplicity and continuous redundancy are two different structural sources of multiplicity of solutions in the feasible domain searched for optimal feasible solutions. As a consequence of their presence, the optimizer may get into a difficult situation because the objective function does not vary over a domain of non-zero measure. Therefore, it is worth analyzing

Abbreviations: BDIMR, binarily decreased and ideal MINLP representation; BGR, basic GDP representation; BMIMR, binarily minimal and ideal MINLP representation; BMR, basic MINLP representation; BMMR, binarily minimal MINLP representation; CMR, conventional MINLP representation; CNF, conjuctive normal form; DNF, disjunctive normal form; GDP, generalized disjunctive programming; IMR, ideal MINLP representation; ILP, integer linear programming; LP, linear programming; MILP, mixed integer linear programming; MINLP, mixed integer non-linear programming; MR, MINLP representation; NLP, non-linear programming; NSXMR, non-considered structures excluded MINLP representation; SEN, state-equipment network; STN, state-task network

* Corresponding author. Tel.: +36 1 463 1189; fax: +36 1 463 3197.
E-mail address: ufo@mail.bme.hu (E. Rev).

0098-1354/$ – see front matter
doi:10.1016/j.compchemeng.2005.07.008

Nomenclature

Parameters

k	maximum number of membrane sections in Example 2
k	maximum number of membrane modules in a section in Example 2
ε	small positive value

Variables

c	cost
cost	cost
Q	integer variable in BMMR
UNITS	number of membrane modules in a section
x	continuous variable
y	binary variable
$\hat{y}$	binary variable in BMMR
z	logical variable

Sets and regions

I0	set of indices for which $\hat{y}_i = 0$
I1	set of indices for which $\hat{y}_i = 1$
R	set of graphs
X	region of continuous variables
Y	region of binary variables

Superscript

UP	upper bound

Subscripts

a	membrane section in Example 2
b	membrane module in a section in Example 2
fix	fix
im	(unit) not present (‘missing’)
in	inlet
ip	(unit) present
j	considered graph
m	unit
out	outlet
var	variable

their sources and looking for ways to avoid these phenomena or at least decrease their effect. The sources of structural multiplicity and redundancy are discussed in Rev, Farkas, and Lelkes (submitted for publication). In that article, an exact definition of process flowsheet superstructure is given, explained, visualized and demonstrated with examples. The phenomena called structural multiplicity and by-pass redundancy are defined.

R-graphs are introduced and used for representing process flowsheets. The process flowsheet structure is defined as a class of isomorphic *R*-graphs. One of the main conclusions of that paper is the following: if each of two considered structures contain a substructure not being simultaneously a substructure of the other one, as is usually the case, then their common superstructure necessarily contains non-considered structures as well. Therefore, an ideal superstructure, representing the considered structures only, usually cannot be constructed.

The second step of the process synthesis activity is constructing a quantitative model of the superstructure and its optimization. The optimization model may be a mixed integer non-linear programming (MINLP) problem or may involve logical constraints as well. Structural multiplicity has a significant effect on solving the MINLP model. On the other hand, the optimization model may also have built-in redundancy that cannot always be directly derived from the multiplicity of the superstructure. Our target in Part I (Farkas, Rev, & Lelkes, 2005) was developing a methodology for constructing MINLP representations (MR-s) of superstructures most appropriate for numerical solution. The essential problems in defining MR are representativeness and uniqueness. Whether an MR represents a superstructure and all its considered substructures should be determinable. Another crucial problem is comparison of two different MINLP representations to decide if some process flowsheets are represented by both of them.

Ambiguousity in defining the variables of the MR was avoided in Part I by first applying a basic GDP representation (BGR), and then constructing a so-called basic MINLP representation (BMR) with standard mapping between them. BMR is defined by transforming the logical relations to algebraic ones. In this way, BMR serves as a reference representation. MR was defined, in Part I, through a fixed form of BMR. Equivalency and representativeness of MR-s in general form can be analyzed by reducing them to their BMRs. BMR can be automatically generated and can serve as a reference representation.

Binary and continuous multiplicity of MR are also defined in Part I. If a supergraph is structurally redundant (i.e. there are isomorphic graphs amongst their subgraphs) then BMR has binary multiplicity. Conversely, the structural redundancy of the supergraph does not follow from the binary multiplicity of its BMR.

In the present part we define MINLP representations possessing beneficial extremal properties, and test them from numerical point of view. First the ideal MR (IMR) and then the binarily minimal MR (BMMR) are defined and tested. Finally, combinations of these extrema, as well as MR-s with decreased structural multiplicity, will be defined and analyzed.

2. Ideality of the MINLP representation

In Section 2 of Part I (Farkas et al., 2005), we outlined the difference between considered and non-considered structures and R-graphs. The first task of the engineer, in the process synthesis, is to assign the considered structures. However, a considered structure may be represented by several isomorphic R-graphs, each being a subgraph of the supergraph. Therefore, the engineer has also to assign the set of considered graphs. The set of considered graphs should not include isomorphic pairs. Unfortunately, representation of non-considered graphs by the MINLP representation is not excluded.

As an analogy with the notion of ideal superstructure, the ideal MINLP representation is an MR which represents the considered graphs only, and no other graphs. This is an MR restricted to a surjection ψ from the feasible domain of MR to the feasible domain of BMR. This mapping ψ contains, as a subrelation, $\pi \subseteq \psi$, where π is a bijection between a subset of the feasible domain of MR and the selected set of graphs, as is explained in Part I.

Here, we also define a measure of non-ideality as follows: let the number of considered subgraphs be $n > 0$; let the number of represented subgraphs be $m \geq n$, then *the measure of non-ideality* is the ratio: $N = (m - n)/n$. This measure reaches very high values if the number of isomorphic subgraphs is great.

Although ideal superstructure does not always exist, moreover, in the important cases it usually does not exist, *ideal MINLP representation always exists*. This existence is shown here by construction. *IMR can always be constructed* as follows.

First, a BGR and the BMR of the superstructure is constructed, as shown in Part I. Here, we remind the reader that once the BGR is constructed, the BMR is unambiguously determined. Then, the criteria of considered structures are expressed in logical relationships. Satisfaction of these relationships excludes all the non-considered graphs; thus it enables just the considered graphs to be present in the solution space. These logical relationships include Boolean relations like ‘and’, ‘or’, ‘not’, ‘imply’, etc., and logical variables. In most of the cases, the applied logical variables are the logical variables of the BGR, expressing the existence of the units. These logical relationships together form a logical truth function, corresponding to Eq. (14) of Part I. Any truth function can be expressed in conjunctive normal form. How it can be done is described, for example, by Raman and Grossmann (1991). Any conjunctive normal form can be transformed into linear equations using binary variables instead of logical ones. By inserting these linear equations into the BMR, the MR formed this way is an IMR because all the non-considered graphs are excluded by the inserted relations.

2.1. Example 1—an ideal MINLP representation

Here, we demonstrate the above procedure for constructing IMR. For this aim, the BGR and BMR representation of Example 1 of Part I, as constructed and presented there, is applied as starting representations; an IMR of the same problem is constructed.

Earlier in the problem statement, the criteria of the considered structures have been defined as:

1. Source stream(s) should exist.
2. Product stream(s) should exist.
3. No other outlet streams than the product stream(s) may be present.
4. Units 2 and 3 should not exist simultaneously.

These criteria can be written via logical variables in the following way:

1. $$z_1 \vee z_4 \quad (1)$$

2. $$z_6 \quad (2)$$

3. $$z_1 \Leftrightarrow z_2 \vee z_3 \quad (3a)$$

$$z_2 \vee z_3 \vee z_4 \Leftrightarrow z_5 \quad (3b)$$

$$z_5 \Leftrightarrow z_6 \quad (3c)$$

4. $$\neg(z_2 \wedge z_3) \quad (4)$$

Eqs. (2) and (3c) are always true, because Units 5 and 6 are permanent units. For the same reason, Eq. (3b) can be written in this form:

$$z_2 \vee z_3 \vee z_4 \quad (5)$$

The conjuctive normal form of these relations is (see Raman & Grossmann, 1991):

1. $$(z_1 \vee z_4)\wedge \quad (6)$$

3. $$(\neg z_1 \vee z_2 \vee z_3)\wedge \quad (7a)$$

$$(z_1 \vee \neg z_2)\wedge \quad (7b)$$

$$(z_1 \vee \neg z_3)\wedge \quad (7c)$$

4. $$(\neg z_2 \vee \neg z_3) \quad (8)$$

This conjuctive normal form can be transformed into linear equations using binary variables:

1. $$y_1 + y_4 \geq 1 \quad (9)$$

3. $$y_1 - y_2 - y_3 \leq 0 \quad (10a)$$

$$-y_1 + y_2 \leq 0 \quad (10b)$$

$$-y_1 + y_3 \leq 0 \quad (10c)$$

4. $$y_2 + y_3 \leq 1 \quad (11)$$

An IMR representation is formed simply by appending the BMR with Eqs. (9)–(11). On the other hand, we are using R-graphs, and it has been shown in Part I that the first three criteria are automatically satisfied using R-graph representation. Therefore, Eqs. (9) and (10a)–(10c) are automatically satisfied, and only the fourth criterion is to be checked. It follows that only Eq. (11) is to be appended to BMR. Then the non-considered graphs are excluded; therefore, the representation becomes ideal.

3. Binarily minimal MINLP representation

The problems including integer or binary variables (ILP/MILP/MINLP) are usually more difficult to solve than the problems without them (LP/NLP). The applied form of the equations and the applied methodology of treating the integer variables (type of relaxation) together determine the effectiveness of the solution algorithm and the scale of the solvable problems. Although a great number of integer variables can be successfully dealt with in special cases, they involve serious difficulty in general case. Usually the difficulty (solution time, for example) of solving the problem drastically increases with the number of integer variables. Therefore, decreasing the number of integer variables has a key role in increasing the scale of the solvable problems and decreasing the solution work.

In principle, all the integer variables can be eliminated by an appropriate transformation of the variables and the equations. For example, any binary variable y can be substituted by a continuous variable x with the following extra constraint: $x(x-1)=0$. In this case, the feasible values of x are 0 and 1. On the other hand, such elimination introduces another type of difficulty, i.e. it is usually unsolvable by the commonly applied solvers. Therefore, compromise is to be found between the number of integer variables and the equation forms applied in the problem formulation. It should also be taken into account that the commonly applied solver algorithms usually enable the binary variables appearing in linear members only in the equations (like in P1 of Part I).

Here, we suggest the compromise that applies the minimum number of binary variables with the constraints that: (i) the structural variants are all distinguished by binary variables and (ii) the linearity of the binary members in the equations is maintained.

From here on, we consider the simple case of using the binary variables to distinguish between the structural variants only. With this simplification, the term 'binary variable' will mean 'binary variable applied to make distinction between different structures'.

An MR is called binarily minimal MINLP representation if it applies a minimum number of binary variables to make distinction between represented subgraphs.

In mathematical sense, it can be formulated as application of the minimum number of binary variables with the condition that an injective (i.e. invertible) mapping can be given from the domain $\mathbf{Y}$ of y (the binary variables) to the set $\mathbf{R}=\{r_1, r_2, \ldots, r_k\}$ of the subgraphs. This condition means that any $r_j \in \mathbf{R}$ graph is assigned to only one value of y. If the MR is BMMR, such a mapping with less number of binary variables cannot be given.

It is useful if one knows how many binary variables are to be minimally used. This can be readily given by knowing how many graphs are to be described. An array of n binary variables can take 2^n different values. This number is to be at least as great as the number k of the R-graphs: $2^n \geq k$. That is, n is the smallest whole number that satisfies $n \geq \log_2 k$. The equality is satisfied only if k is exactly a whole power of 2.

In practice, the above set $\mathbf{R}$ of described graphs is to be the set of considered graphs. If k is a whole power of 2 then the BMMR is necessarily an IMR because no other graph is described than those in $\mathbf{R}$. However, if k is not a whole power of 2 then the number of graphs described by the binary variables can be greater than k. For example, let $k=10$; then $n=4$ (because $2^3=8<10<16=2^4$), and $2^n=2^4=16$. Thus, one to six additional graphs (numbered as 11, 12, ..., 16) could be described by the binary variables depending on if they are part of the feasible region or not. Should the BMMR not be IMR, it can always be made also an IMR by applying extra constraints that exclude the superfluous graphs from the feasible region.

Binarily minimal MINLP representation can always be constructed: here, we prove this theorem by construction. The method presented here for constructing BMMR serves this purpose only, and should not be considered as the only or the best methodology. Different methodologies can be elaborated, and numerical viewpoints can be taken into account, when the engineer construct a proper methodology for the particular shape of the problem at hand. The methodology used here as a proof is that follows:

1. The conjunctive normal form (CNF), mentioned in the section above describing how to construct IMR, is to be converted into disjunctive normal form (DNF) where all the logical variables take place in each member of the expression. This can be accomplished in an automatic procedure (see its application for process synthesis, for example, in Raman & Grossmann, 1991). The DNF consists of a series of *clauses* (i.e. brackets) joined with 'or' relation, the brackets containing logical items, with or without the operator 'not', separated with 'and'. The DNF looks like $[(z_1 \wedge \neg z_2 \wedge z_3 \cdots) \vee (\neg z_1 \wedge z_2 \wedge z_3 \cdots) \vee (z_1 \wedge \neg z_2 \wedge \neg z_3 \cdots) \vee \cdots]$, where the z's, separated by 'and', are the logical variables of the BGR. Each considered graph r_j is assigned the integer index j ($1 \leq j \leq k$). Each of the *clauses* separated by 'or' corresponds to a considered graph r_j (and, in turn, to a distinct structure, see Brendel, Friedler, & Fan, 2000). That is, DNF is given in the form:

$$\bigvee_{j=1,2,\ldots,k} \Big(\bigwedge_{\forall \text{ip present}} z_{\text{ip}} \bigwedge_{\forall \text{im missing}} \neg z_{\text{im}} \Big)_j \qquad (12)$$

where ip are indices of units present in considered graph r_j, whereas im are indices of units not present in considered graph r_j.

2. The above j indexes as whole numbers can be expressed in binary form. The minimum number of necessary binary variables is determined by the maximum of these indexes, denoted by k. This minimum, as we have seen above, is the smallest whole number n that satisfies $n \geq \log_2 k$. Let the binary digits be described by using the binary variables $\hat{y}_i$ ($i = 0, 1, 2, \ldots, n-1$) so that the index j is expressed as

$$j = 1 + \sum_{i=0}^{n-1} \hat{y}_i 2^i \quad (13)$$

This expression of a particular number j defines an index set I1 of indices i for which $\hat{y}_i = 1$, and another index set I0 of indices i for which $\hat{y}_i = 0$. This definition of index sets depends on the actual value of j. Therefore, these index sets are given as functions of j:

$$\begin{aligned} \mathrm{I1}(j) &= \{i | \hat{y}_i = 1, \quad \text{in Eq. (13)}\} \\ \mathrm{I0}(j) &= \{i | \hat{y}_i = 0, \quad \text{in Eq. (13)}\} \end{aligned} \quad (14)$$

3. Then, a set of integer variables Q_j ($j = 1, 2, \ldots, k$) is defined as:

$$Q_j = \sum_{i \in \mathrm{I1}(j)} (1 - \hat{y}_i) + \sum_{i \in \mathrm{I0}(j)} \hat{y}_i \quad (15)$$

If the actual value of $\hat{y}$ substituted into Eq. (13) gives the value j then $Q_j = 0$; otherwise $Q_j \geq 1$.

4. In the next step, the BMR is transformed into BMMR using the above defined $\hat{y}$ and Q_j variables instead of y. The BMR is formed using Big M technique. The Big M equations of the BMR contains members multiplied by factors of y_i or $(1 - y_i)$, see Eqs. (19a)–(29) of Part I in general, and Eqs. (32)–(38) of Example 1 in Part I. These latter equations are listed in Appendix A.

First, the k equations from Eq. (15) ($j = 1, 2, \ldots, k$) are introduced.

Then, for each unit m, it can be easily decided if it is contained by graph j, using the decomposition of the index set of units (or their logical 'presence' variables z_m) to ip and im, according to Eq. (12).

For each unit m, and for each equation of unit m that *does* contain a factor y_m, this equation is used as many times as many considered graphs contain unit m. In each of these copies the factor y_m is substituted by a unique Q_j, where j is the index of a considered graph that contains unit m. That is, each such j is applied by turn.

For each unit m, and for each equation of unit m that contains a factor $(1 - y_m)$, this equation is used as many times as many considered graphs *do not* contain unit m. In each of these copies the factor $(1 - y_m)$ is substituted by a unique Q_j, where j is the index of a considered graph that *does not* contain unit m. That is, each such j is applied by turn.

That is, for each unit m, k variants of each equation are, in principle, formed. In practice, many of these newly formed equations are redundant, therefore may be omitted.

At this point, i.e. by completing step 4, BMMR is formed. This form of BMMR has the merit that all the equations containing binary variables are linear; i.e. no non-linear equation is added to the original form of BMR.

3.1. Example 1—a binarily minimal MINLP representation

Here, we demonstrate the above procedure for constructing the BMMR of the same example to which the basic GDP representation and the basic MINLP representation was already shown in Part I.

The disjunctive normal form can be accomplished from conjuctive normal form (Eqs. (16)–(18)) in an automatic procedure. This is Eq. (16). Each of the clauses corresponds to a considered graph. For example, the z_3 logical variable is false in the first clause; all the other logical variables are true. Thus, this clause corresponds to a graph in which all the units exist except Unit 3. This graph is shown in Fig. 11a of Part I. There are five considered graphs, these are presented in Fig. 11a–e of Part I, and all these graphs are denoted by one clause in DNF. The considered graphs are assigned the integer index j. It is shown in Eq. (16).

	R-graph	j	
$(z_1 \wedge z_2 \wedge \neg z_3 \wedge z_4 \wedge z_5 \wedge z_6)\vee$	Fig.11a	1	
$\vee(z_1 \wedge z_2 \wedge \neg z_3 \wedge z_4 \wedge z_5 \wedge z_6)\vee$	Fig.11b	2	
$\vee(z_1 \wedge \neg z_2 \wedge z_3 \wedge z_4 \wedge z_5 \wedge z_6)\vee$	Fig.11c	3	(16)
$\vee(z_1 \wedge \neg z_2 \wedge z_3 \wedge \neg z_4 \wedge z_5 \wedge z_6)\vee$	Fig.11d	4	
$\vee(\neg z_1 \wedge \neg z_2 \wedge \neg z_3 \wedge z_4 \wedge z_5 \wedge z_6)\vee$	Fig.11e	5	

There are five considered graphs, therefore $3 \geq \log_2 5$ binary variables are necessary to generate the binarily minimal MINLP representation. Using the three new binary variables, any integer variable j can be expressed according to Eq. (17):

$$j = 1 + \hat{y}_1 + 2 \cdot \hat{y}_2 + 4 \cdot \hat{y}_3 \quad (17)$$

Then, the set of integer variables Q_j can be defined as in Eq. (18):

$$\begin{aligned} Q_1 &= \hat{y}_1 + \hat{y}_2 + \hat{y}_3 \\ Q_2 &= (1 - \hat{y}_1) + \hat{y}_2 + \hat{y}_3 \\ Q_3 &= \hat{y}_1 + (1 - \hat{y}_2) + \hat{y}_3 \\ Q_4 &= (1 - \hat{y}_1) + (1 - \hat{y}_2) + \hat{y}_3 \\ Q_5 &= \hat{y}_1 + \hat{y}_2 + (1 - \hat{y}_3) \end{aligned} \quad (18)$$

For example, the graph shown in Fig. 11d of Part I is represented by $j = 4$ via Eqs. (16)–(18). According to Eq. (17), variable j can take the value 4 if $\hat{y} = [1, 1, 0]$. It means that $Q_4 = 0$ and all the other Q_j-s take positive integer value ($Q_1 = 2$; $Q_2 = 1$; $Q_3 = 1$; $Q_5 = 3$).

In the next step, the BMR is transformed into BMMR. Only those equations are changed which contain binary variables (Eqs. (32)–(38) of Part I). The transformed equations of the input streams are given in Eqs. (19a)–(19d).

$$0 \le x_{2,\mathrm{in}} \le 2.04 Q_j, \quad j = 3\text{–}5 \tag{19a}$$

$$-x_{2,\mathrm{in}} \le 2.04 Q_j - \varepsilon, \quad j = 1, 2 \tag{19b}$$

$$0 \le x_{3,\mathrm{in}} \le 1.53 Q_j, \quad j = 1, 2, 5 \tag{19c}$$

$$-x_{3,\mathrm{in}} \le 1.53 Q_j - \varepsilon, \quad j = 3, 4 \tag{19d}$$

$x_{2,\mathrm{in}}$ takes positive value if Unit 2 exists. According to Eq. (16), Unit 2 exists if $j = 1$ or 2, and does not exist if $j = 3$, 4 or 5. Consider, for example, the case where $j = 3$ and Unit 2 does not exist. $Q_3 = 0$ according to Eq. (18), and the other Q_j ($j \ne 3$) variables take positive integer value. The three different equations denoted by Eq. (19a) are detailed in Eqs. (20a)–(20c). Variable $x_{2,\mathrm{in}}$ is forced in Eq. (20a) to take the value 0. $Q_4 \ge 1$ and $Q_5 \ge 1$ in the two other Eqs. (20b) and (20c); therefore $0 \le x_{2,\mathrm{in}} \le M$, where M is a number not smaller than the upper bound (2.04) of $x_{2,\mathrm{in}}$. The two equations of Eq. (19b) are detailed in Eqs. (21a) and (21b): If $j = 3$ then $Q_1 \ge 1$ and $Q_2 \ge 1$; therefore, these equations are satisfied by the assignment $x_{2,\mathrm{in}} = 0$.

$$0 \le x_{2,\mathrm{in}} \le 2.04 Q_3 \tag{20a}$$

$$0 \le x_{2,\mathrm{in}} \le 2.04 Q_4 \tag{20b}$$

$$0 \le x_{2,\mathrm{in}} \le 2.04 Q_5 \tag{20c}$$

$$-x_{2,\mathrm{in}} \le 2.04 Q_1 - \varepsilon \tag{21a}$$

$$-x_{2,\mathrm{in}} \le 2.04 Q_2 - \varepsilon \tag{21b}$$

Another example is the case where $j = 1$ and Unit 2 exists. $Q_1 = 0$ and $Q_{j \ne 1} \ge 1$ in this case. Variable $x_{2,\mathrm{in}}$ has to take a positive value not smaller then ε, according to Eq. (21a). But none of the other Eqs. (20a)–(20c) and (21b) effects the value of $x_{2,\mathrm{in}}$; thus, $x_{2,\mathrm{in}}$ can take any value in interval [ε, $(x_{2,\mathrm{in}})^{\mathrm{UP}} = 2.04$].

Eqs. (19c) and (19d) work similarly. Variable $x_{3,\mathrm{in}}$ is positive if Unit 3 exists and if $j = 3$ or 4, according to Eq. (16). If $j = 1$, 2 or 5 then Unit 3 does not exist, and Eq. (19c) forces $x_{3,\mathrm{in}}$ to take value 0. If $j = 3$ or 4 then Unit 3 exists and $x_{3,\mathrm{in}}$ can take any value in interval [ε, $(x_{2,\mathrm{in}})^{\mathrm{UP}} = 1.53$].

Similar equations can be written for the output streams (Eq. (22)), for the fix and variable costs (Eqs. (23) and (24)), and for the functions of the output streams and costs (Eqs. (25)–(27)).

$$\begin{aligned}
&0 \le x_{1,\mathrm{out}} \le 3.57 Q_j, \quad j = 5\\
&-x_{1,\mathrm{out}} \le 3.57 Q_j - \varepsilon, \quad j = 1\text{–}4\\
&0 \le x_{2,\mathrm{out}} \le 1.11 Q_j, \quad j = 3\text{–}5\\
&-x_{2,\mathrm{out}} \le 1.11 Q_j - \varepsilon, \quad j = 1, 2\\
&0 \le x_{3,\mathrm{out}} \le 1.11 Q_j, \quad j = 1, 2, 5\\
&-x_{3,\mathrm{out}} \le 1.11 Q_j - \varepsilon, \quad j = 3, 4\\
&0 \le x_{4,\mathrm{out}} \le 1.11 Q_j, \quad j = 2, 4\\
&-x_{4,\mathrm{out}} \le 1.11 Q_j - \varepsilon, \quad j = 1, 3, 5
\end{aligned} \tag{22}$$

$$\begin{aligned}
&0 \le c_{\mathrm{fix},1} \le 0 Q_j, \quad j = 5\\
&0 \le c_{\mathrm{fix},2} \le 1 Q_j, \quad j = 3\text{–}5\\
&0 \le c_{\mathrm{fix},3} \le 1.5 Q_j, \quad j = 1, 2, 5\\
&0 \le c_{\mathrm{fix},4} \le 0 Q_j, \quad j = 2, 4
\end{aligned} \tag{23}$$

$$\begin{aligned}
&0 Q_j \le c_{\mathrm{var},1} \le 6.42 Q_j, \quad j = 5\\
&0 Q_j \le c_{\mathrm{var},2} \le 5 Q_j, \quad j = 3\text{–}5\\
&0 Q_j \le c_{\mathrm{var},3} \le 2.31 Q_j, \quad j = 1, 2, 5\\
&0 Q_j \le c_{\mathrm{var},4} \le 7.78 Q_j, \quad j = 2, 4
\end{aligned} \tag{24}$$

$$\begin{aligned}
&-1.11 Q_j \le x_{2,\mathrm{out}} - \ln(x_{2,\mathrm{in}} + 1) \le 1.11 Q_{j'}, \quad j = 3\text{–}5;\ j' = 1, 2\\
&-1.11 Q_j \le -x_{2,\mathrm{out}} + \ln(x_{2,\mathrm{in}} + 1) \le 1.11 Q_{j'}, \quad j = 3\text{–}5;\ j' = 1, 2\\
&-1.11 Q_j \le x_{3,\mathrm{out}} - 1.2\ln(x_{3,\mathrm{in}} + 1) \le 1.11 Q_{j'}, \quad j = 1, 2, 5;\ j' = 3, 4\\
&-1.11 Q_j \le -x_{3,\mathrm{out}} + 1.2\ln(x_{3,\mathrm{in}} + 1) \le 1.11 Q_{j'}, \quad j = 1, 2, 5;\ j' = 3, 4
\end{aligned} \tag{25}$$

$$\begin{aligned}
&0 Q_j \le c_{\mathrm{fix},1} - 0 \le 0 Q_j, \quad j = 1\text{–}4\\
&-1 Q_j \le c_{\mathrm{fix},2} - 1 \le 0 Q_j, \quad j = 1, 2\\
&-1.5 Q_j \le c_{\mathrm{fix},3} - 1.5 \le 0 Q_j, \quad j = 3, 4\\
&0 Q_j \le c_{\mathrm{fix},4} - 0 \le 0 Q_j, \quad j = 1, 3, 5
\end{aligned} \tag{26}$$

$$\begin{aligned}
&-6.42 Q_j \le c_{\mathrm{var},1} - 1.8 x_{1,\mathrm{out}} \le 6.42 Q_j, \quad j = 1\text{–}4\\
&-5 Q_j \le c_{\mathrm{var},2} - x_{2,\mathrm{out}} \le 5 Q_j, \quad j = 1, 2\\
&-2.31 Q_j \le c_{\mathrm{var},3} - 1.2 x_{3,\mathrm{out}} \le 2.31 Q_j, \quad j = 3, 4\\
&-7.78 Q_j \le c_{\mathrm{var},4} - 7 x_{4,\mathrm{out}} \le 7.78 Q_j, \quad j = 1, 3, 5
\end{aligned} \tag{27}$$

The transformation is the same in case of all the equations. If an equation contains the factor y_m, it is substituted by a set of equations which contains unique Q_j, where j is the index of a considered graph that *does not* contain unit m. Similarly, equations containing the factor $(1 - y_m)$ are substituted by sets of equations, which contain unique Q_j, where j is the index of a considered graph that contains unit m.

As a result of the above transformation, the MINLP representation becomes binarily minimal because it uses the minimal number of binary variables. But it is still not an ideal representation. Combination of three binary variables can take $2^3 = 8$ different values. Thus, variable j can also take eight different values, according to Eq. (17), whereas there are only five considered graphs. In the cases of $\hat{y} = [1, 0, 1]$, [0, 1, 1] and [1, 1, 1], variable j takes value $j = 6$, 7 and 8, respectively. But these values of j do not denote considered graphs, therefore these values have to be excluded from the representation. This can be done by inserting integer cuts in

Table 1
Solution times in Example 1

Representation	Objective value	Number of iterations	Solution time (s)	NLP (s)	MILP (s)	NLP/it. (s)	MILP/it. (s)
MR of Kocis	−1.923	8	0.98	0.61	0.37	0.076	0.046
MR	−1.923	16	2.37	1.18	1.19	0.074	0.074
IMR	−1.923	12	1.70	0.86	0.84	0.072	0.070
BMIMR	−1.923	5	0.92	0.44	0.48	0.088	0.096

the form of Eq. (28):

$$\begin{aligned} &\hat{y}_1 - \hat{y}_2 + \hat{y}_3 \le 1 \\ &\hat{y}_1 + \hat{y}_2 + \hat{y}_3 \le 2 \\ &-\hat{y}_1 + \hat{y}_2 + \hat{y}_3 \le 1 \end{aligned} \tag{28}$$

After adding Eq. (28) to the representation, it represents the considered graphs only, thus it is ideal. Since, in the same time, it uses the minimal number of binary variables, this representation is binarily minimal *and* ideal. Therefore, it is abbreviated as BMIMR.

The above procedure is just one possible method for generating BMIMR; there may be several others. The main advantage of this method is that no more non-linear equations are added to the representation. On the other hand, relaxation of the Big M equations become worse in this way. However, decrease of the total running time is expected because of decreasing the number of the main (outer) iterations.

4. Example 1—computational results

Here, we compare the different models outlined above and in Part I according to our computational results with the same example problem hitherto discussed. For this aim, the synthesis problem of Kocis and Grossmann (1987) is studied.

First, the MINLP representation originally given by Kocis and Grossmann ("MR of Kocis") was solved, for comparing the running time. Second, another MINLP representation (MR) was constructed by first constructing the BMR of the problem and then excluding the redundant and unnecessary equations. Third, the non-considered graphs were excluded from this MR by inserting logical constraints; so that it became ideal ("IMR"). Finally, the binarily minimal and, in the same time, ideal MINLP representation ("BMIMR") was generated via using minimal number of binary variables.

These four representations were solved on a Sun Sparc workstation, using GAMS DICOPT++ solver (Brooke, Kendrick, & Meeraus, 1988). All the binary variables were assigned the initial value 0.5 in each case. The stop criterion was set to "STOP 0" because of the presence of non-linear equations. This parameter value results in stopping the iteration only if the mixed integer linear programming (MILP) subproblem becomes infeasible. The solution times and results are collected in Table 1.

The second column of Table 1 shows the optimal value of the objective function. The number of iterations shows how many main (outer) iterations were done. The solution time is given in CPU sec, and is also broken down to NLP and MILP subproblems. The average NLP and MILP CPU time, per iteration, is given in the last two columns, for comparison.

The same optimum was found in each case. This solution is shown in Fig. 1. This graph is represented by $j = 4$ in our BMIMR, as detailed in an earlier section. Sixteen iterations were needed, using MR, to consider all the feasible MILP subproblems. The number of iterations decreased to 12 as a results of using IMR instead (the non-considered graphs were excluded). The solution time of the NLP and MILP subproblems, and therefore the total solution time as well, decreased by about 28–30%. Using minimal number of binary variables (BMIMR) resulted in decreasing the solution time by 46%. In the same time, a small increase in the solution time per iterations is observed. This effect may be caused by three factors: (1) there are more equations in BMIMR than in IMR; therefore, the NLP-problems became greater; (2) the relaxation is not so good in this case; (3) the number of equations containing binary variables also increased; therefore, the MILP subproblems became more complex. On the other hand, only six iterations were necessary in this case to investigate all the feasible MINLP subproblems, thanks to having less number of binary variables. This is, perhaps, why the total solution time decreased.

The MINLP representation of Kocis and Grossmann (1987) is binarily minimal because it uses no more than three binary variables. We suspect, this is the reason why applying MR of Kocis involved faster solution than MR and IMR did.

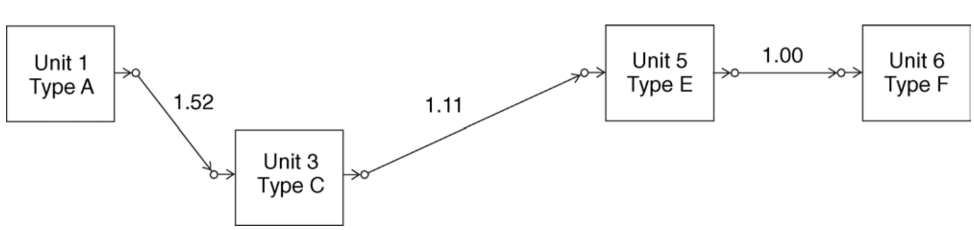


Fig. 1. The optimal solution of Example 1.

On the other hand, it is not ideal according to our criterion because it represents graphs including Units 2 and 3 simultaneously, which are non-considered graphs in our example. The most probable reasons that resulted in 6% faster solution of our BMIMR than MR of Kocis, even against the somewhat worse relaxation of the MILP subproblems, are the following two properties: (1) BMIMR is, according to its name, ideal representation, i.e. all the non-considered graphs are excluded from the search space. (2) Only some of the variables are given upper bounds in MR of Kocis.

In this example, we applied just one method for constructing IMR and BMIMR. The same methodology can be applied to any MINLP problem, but not always will, perhaps, improve the numerical properties of the solution procedure. The number of necessary iterations is decreased to its third in one case, and the solution time is decreased to 39%, even for this rather small example, by idealizing the representation.

5. Example 2—pervaporation system

Here we discuss *the membrane subsystem* of an industrial distillation and membrane hybrid system of ethanol dehydration, presented by Szitkai, Lelkes, Rev, and Fonyo (2002). Our aim with this example is to show how idealization of the MINLP representation and reduction of the number of binary variables result in both faster solution and essential increase in the problem scale solvable with the commercial MINLP solvers.

The ethanol dehydration system has two main parts (Fig. 2): a distillation column and a pervaporation system. The top product of the column is ethanol–water mixture of near azeotropic composition, and is fed to the first section of a membrane train. The membrane train is a series of several sections each containing a set of parallel connected membrane modules having the same inlet in that section. The retentate streams of the membrane modules are collected; a part is led away as product, whereas the other part is fed to the next section of membranes through a heat exchanger. The permeate streams of the membranes are also collected and mixed to the feed of the distillation column.

In this example, we consider the membrane subsystem alone, with given distillate properties. The flowrate of the feed to the membrane subsystem is 50 kg/h, and it contains 10.05 mol% of water. The target is to reach 4 mol% of water in the final retentate. In this article, we simplify the problem originally presented by Szitkai et al. (2002) so that here we consider the flowrates and the concentrations only as the parameters of the flows. The main difference is in the objective function (Eq. (29)).

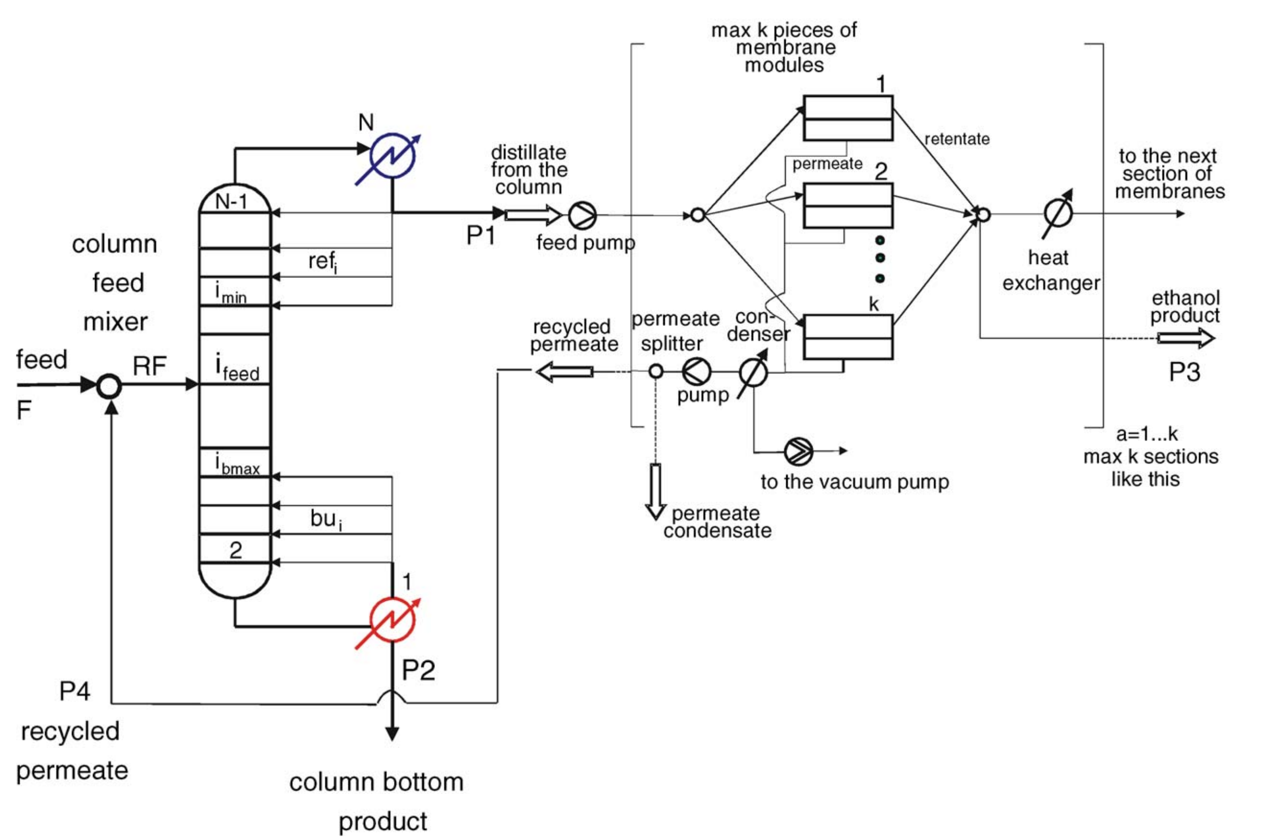


Fig. 2. Superstructure of the ethanol dehydration system.

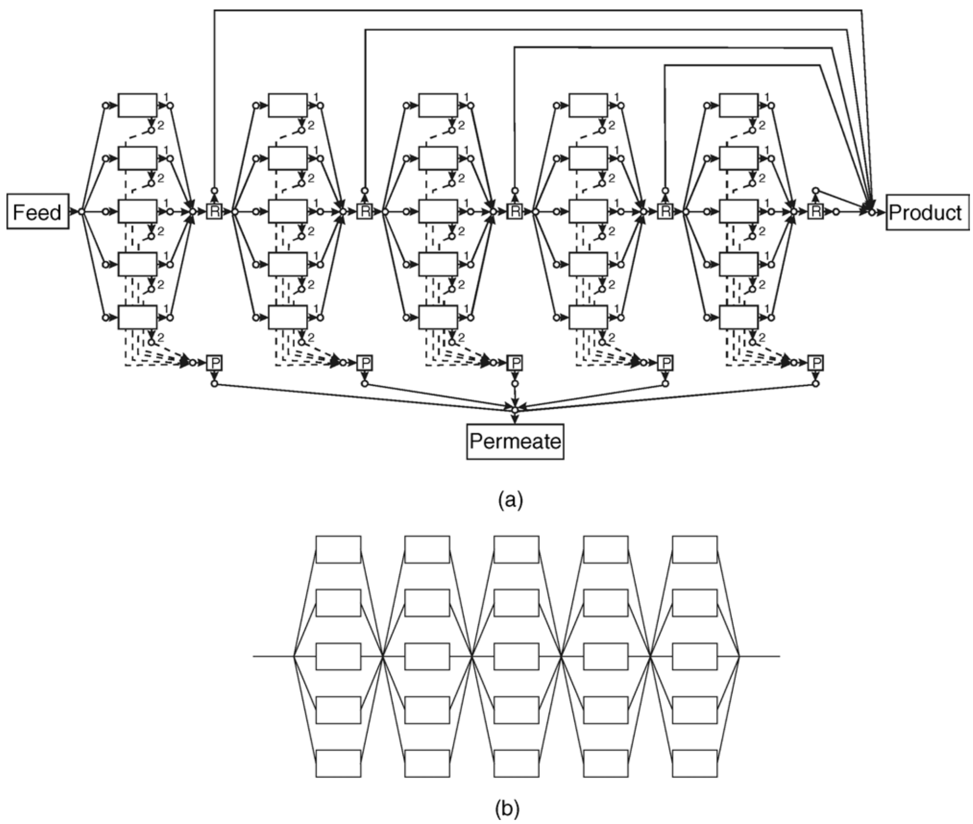


Fig. 3. (a) R-graph of the superstructure in Example 2. (b) Simplified visualization of R-graph representation.

$$\mathrm{cost}_{\mathrm{var}} = \frac{\text{US\$ 775 per year}}{3} \times \sum_{a=1}^{k} \mathrm{UNITS}_a$$
$$\mathrm{cost}_{\mathrm{fix}} = \text{US\$ 161.6108 per year} \times \sum_{a=1}^{k} \mathrm{UNITS}_a \quad (29)$$
$$\text{total cost} = \mathrm{cost}_{\mathrm{var}} + \mathrm{cost}_{\mathrm{fix}}$$

where $\mathrm{cost}_{\mathrm{var}}$ is the variable cost, $\mathrm{cost}_{\mathrm{fix}}$ the fixed cost and UNITS_a is the number of membrane modules in section a.

The first step in solving this problem is defining a superstructure. For this aim, we constraint the membrane system in a "square" shape of scale k. The system consists of maximum k number of sections, and in each section maximum the same k number of membrane modules can be built in. The membrane modules are modelled as if they were all built up from three membrane cells of 1/3 m^3 surface.

The R-graph representation of the superstructure is generated as follows. The supergraph for five membrane sections and five membrane units in each section (a 5×5 system) is shown in Fig. 3a. There is a source unit 'Feed', and there are two sink units ('Product' and 'Permeate') in the supergraph. Each membrane module is modelled as a unit with one input port and two output ports. Output port 1 is used for releasing the retentate, and output port 2 is for the permeate.

Rev et al. (submitted for publication) mentioned that the use of mixer and splitter units is not recommended, because this may lead to redundancy. In order to avoid constructing an unnecessarily complicated supergraph, however, we introduced here mixers and splitters in such a special arrangement that does not introduce by-pass redundancy. This is achieved by leading all the by-pass lines to the final product sink unit 'Product'. After each section there is a mixer/splitter unit, denoted by R, which collects the retentate from the modules in that section, and splits it between the modules of the next section and sink unit 'Product'. Similarly, there is a mixer (denoted by P), which collects the permeate of each section, and leads it to the sink unit 'Permeate'. The retentate mixer–splitter unit after the last section is inserted in order to have the same model for each section. The ratio of one of its output streams to its input stream will be fixed to a constant in each MINLP representation.

A simplified visualization of the R-graphs is applied from here on. The membrane units are denoted by rectangles, the mixer and mixer–splitter units ('P' and 'R'), as well as the source unit 'Feed' and the sink units 'Permeate' and 'Retentate' are omitted. The corresponding streams are also omitted, and only the main streams are shown. The direction of the streams are also omitted, the edges are always directed from left to right. Shaded rectangles represent built in membrane modules, whereas white rectangles denote empty places of

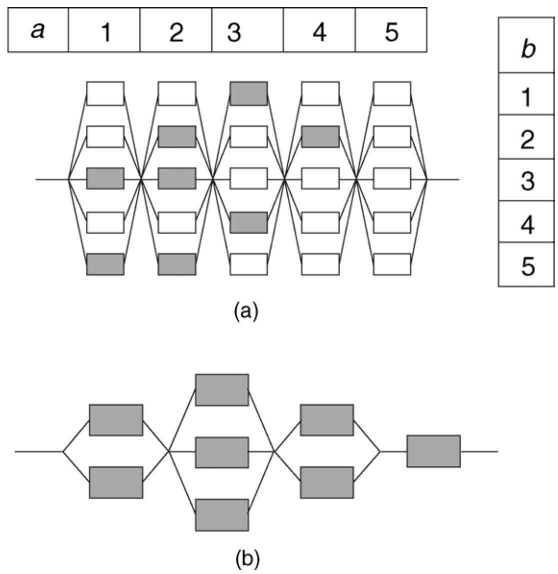


Fig. 4. (a) Graph of a non-considered structure. (b) A non-considered structure.

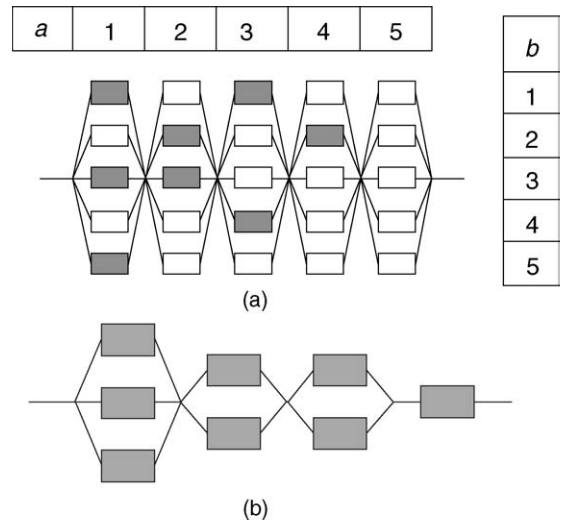


Fig. 5. (a) Graph of a considered structure. (b) A considered structure.

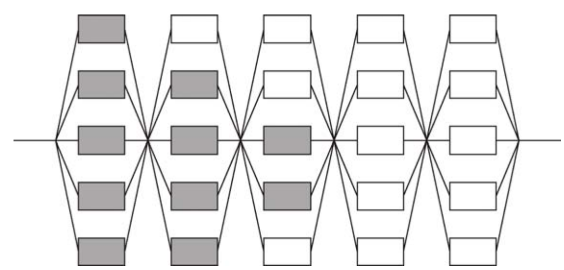


Fig. 6. Optimal solution with CMR, $k = 5$.

modules. For example, a simplified visualization of the 5×5 supergraph is shown in Fig. 3b.

When writing the term 'graph' or 'structure', we always think the full graph version. For example, when calling Fig. 3b 'graph' we mean the graph shown in Fig. 3a.

According to engineering experiences, the number of parallel modules need not be increased as the material flows downstream. Such a structure would not be optimal. Therefore, only those structures will be considered that are characterized with non-increasing number of membrane modules in consecutive membrane sections. For example, the network in Fig. 4a is non-considered structure (Fig. 4b) because there are more membrane modules in the second section than in the first section. This structure includes eight membrane modules. A considered structure (Fig. 5b) with the same number of membrane modules (i.e. eight) is shown in Fig. 5a.

5.1. Conventional MINLP representation

Here, we directly construct an MINLP representation of Example 2 according to the conventional approach.

Binary variables are applied to represent the existence of the membrane modules. If $y_{a,b} = 1$ then place b of section a is occupied by a membrane module, i.e. that module exists, whereas it is not occupied if $y_{a,b} = 0$. (Here, a and b may run from 1 to k, independently.) The same approach is applied in the BMR representation of R-graphs. However, the MINLP representation used here is not BMR because the redundant and unnecessary equations are excluded. Therefore, this MR is called conventional MINLP representation, and denoted by CMR

GAMS DICOPT++ with CONOPT NLP-solver and OSL MINLP-solver on a Sun Sparc station were used for solving this CMR. The initial value for each binary variable was 0.5. Maximum 50 main iterations, and in all the iterations maximum 50,000 iteration steps, were allowed. The CMR has been solved first with $k = 5$, and then with increased k. The solution times and results are collected in Table 2; the optimal solutions are shown in Figs. 6–8.

The problem size k, as an upper limit of both the consecutive membrane sections and of the number of parallel modules in each section, is shown in the first column. The 'cost' column shows the total cost found in the optimal solution. The solution time is broken down into NLP and MILP subproblem

Table 2
Results of CMR

k	Cost (US$/year)	Number of iterations	Solution time (s)	NLP (s)	MILP (s)	NLP/it. (s)	MILP/it. (s)	Non-ideality
5	4619	7	8.42	0.39	8.03	0.056	1.147	11515
6	4619	3	10.77	0.34	10.43	0.113	3.477	207490
7	4619	3	39.65	0.47	39.18	0.157	13.060	3784425
8	–	–	–	–	–	–	–	69023979

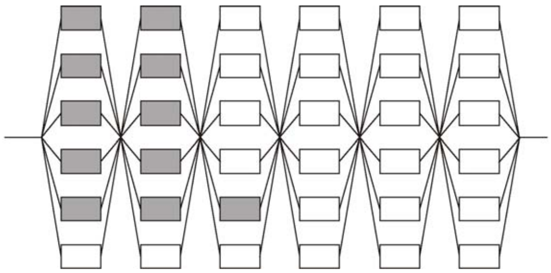


Fig. 7. Optimal solution with CMR, $k = 6$.

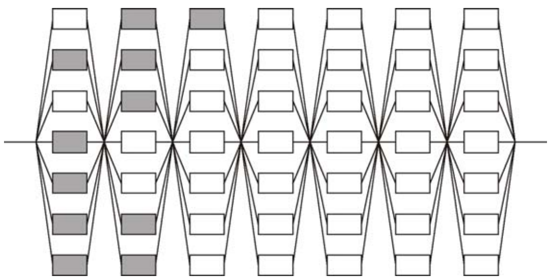


Fig. 8. Optimal solution with CMR, $k = 7$.

time; these subtime values are also given as average values related to iterations. The non-ideality of the representation, i.e. how many non-considered graphs are also represented by the CMR, is given in the last column. The solution consists of 11 membrane modules in each case, as is shown in Figs. 6–8, corresponding to the lines of Table 2. For $k = 8$, and for bigger problems, the solver does not find integer solution within the iteration limit.

The non-ideality of the representations are incredible high, as is seen in the last column. There are two reasons of this great non-ideality:

1. There is not any equation in CMR that would force the solution to satisfy the criterion of considered structures, i.e. the non-increasing number of membrane modules in consecutive membrane sections. On the other hand, each solution in Figs. 6–8 agrees this assumption; therefore, this criterion seems to be right.
2. CMR represents incredible high number of isomorphic graphs, and most of them are non-considered graphs. Consider, for example, a system with maximum five sections, and maximum five modules in each section ($k = 5$). If there is only one membrane module in the actual structure, and if the criterion of non-increasing number of membrane modules in consecutive membrane sections *is satisfied* then the structure can be represented with five isomorphic graphs because that single module can be placed in any of the possible places in the first section. This single unit cannot be put in the second or a later section, because in this case there were no route from the feed to that unit. Two of these five graphs are shown in Fig. 9. These graphs represent the same structure. It follows that the structural multiplicity of the structure consisting of one membrane is five in CMR, in case of $k = 5$. But only one of these isomorphic graphs is a considered graph, the others are non-considered ones. As the CMR represents all the feasible graphs, it represents the non-considered graphs as well; therefore, CMR is not an ideal representation.

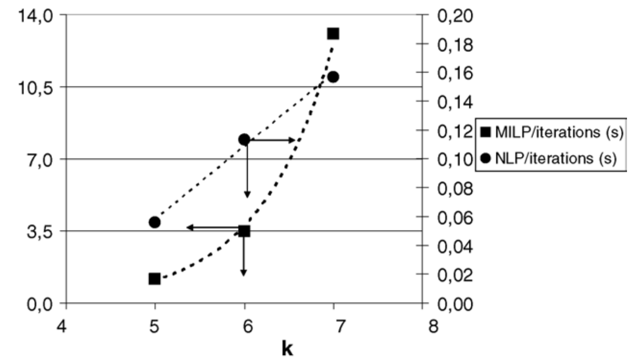


Fig. 10. Solution times per iterations in CMR.

The number of represented graphs (m) and that of the considered graphs (n) are needed to calculate the non-ideality $N = (m - n)/n$ of a representation. Determining them analytically is a rather complicated task. We are content with approximating non-ideality measures for comparing the problems according to this viewpoint. Therefore, the non-ideality is estimated for $k = 2$–4, and a trendline is set in logarithmic plot. The non-ideality of the representation can be estimated for bigger problems, as well, by applying approximating equations of this trendline (Farkas, 2001).

The NLP solution time per iterations increases with the size of the problem (k) linearly, whereas the MILP solution time per iterations increases exponentially, as is shown in Fig. 10.

The linear increase of the NLP solution time per iterations is caused by the increase of the size of the problem and the equation system. The non-ideality (see Table 2) and the

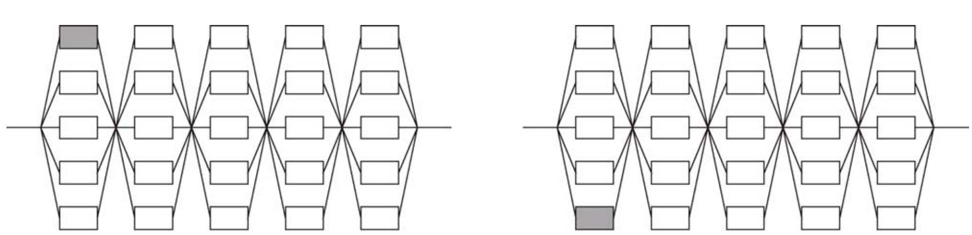


Fig. 9. Isomorphic graphs.

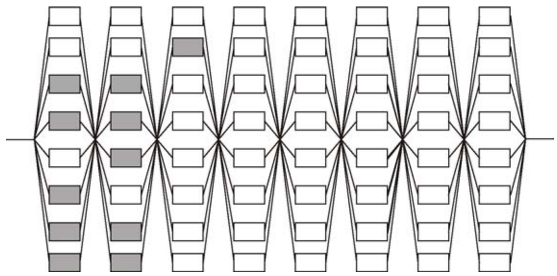


Fig. 11. Optimal solution with NSXMR, $k = 8$.

number of binary variables increase exponentially with the size of the problem; therefore, the search space of the MILP problem also increases exponentially. That is the reason of the exponential increase of the MILP solution time per iterations. The non-ideality increases drastically with increasing problem size, and this is why the problem becomes insolvable in case of 8 × 8 membrane systems and above.

5.2. Exclusion of the non-considered structures

Non-ideality can be decreased by excluding those graphs, from the MINLP representation, that represent non-considered structures. Considered structures are the structures with non-increasing number of membrane modules in consecutive membrane sections (refer to Figs. 4 and 5, and the corresponding notes in the text). The search space can be decreased by excluding some of, or all, the, graphs representing non-considered structures.

In order to exclude the graphs of non-considered structures from the representation, Eq. (30) is added to CMR. This equation forces the solution to satisfy the criterion of non-increasing number of membrane modules in consecutive membrane sections. This representation is called non-considered structures excluded MINLP representation (NSXMR).

$$\sum_{b=1}^{k} y_{a,b} \geq \sum_{b=1}^{k} y_{a+1,b}, \quad a = 1, 2, \ldots, k-1 \tag{30}$$

Problems of greater scale are expected to be solvable by applying MR with decreased non-ideality. Therefore, case $k = 8$, which was infeasible by CMR, was solved using NSXMR. Solution time values are collected in Table 3; the optimal solution is shown in Fig. 11. The non-ideality of the representation was estimated in the same way as in the case of CMR.

The non-ideality of NSXMR with $k = 8$ is smaller by two orders of magnitude, comparing to CMR. Even more, it is smaller than the non-ideality of CMR in case of $k = 7$. That is why the 8 × 8 problem is solvable using NSXMR.

Non-ideality of NSXMR with $k = 9$ is two times greater than that of CMR with $k = 7$. CMR with $k = 7$ contains 109 equations and 147 variables, whereas NSXMR with $k = 9$ consists of 146 equations and 205 variables; i.e. their solutions are approximately of the same difficulty. This is why the 9 × 9 problem is not solvable using NSXMR.

5.3. Ideal MINLP representation

The non-ideality can be further decreased by adding more inequality equations to NSXMR in order to exclude all the non-considered isomorphic graphs from the representation. Isomorphic graphs represent the same structure; therefore, only one of them is considered, whereas the other graphs are non-considered ones. By excluding these non-considered graphs from the MINLP representation, it becomes ideal.

Non-considered graphs can be excluded from the representation by adding the criterion that the membrane modules should fill in the free places of the actual section in a queue. In other words: if there are h membrane modules in a section, these should be put to the first h places. This criterion can be expressed in algebraic form with the help of the binary variables: Each binary variable of a module in a section should not be less than the binary variable of the next module in the same section. This criterion is expressed with Eq. (31).

$$y_{a,b} \geq y_{a,b+1}, \quad a = 1, 2, \ldots, k; \quad b = 1, 2, \ldots, k-1 \tag{31}$$

Consider, for illustration, the two isomorphic graphs shown in Fig. 12; these graphs represent the same structure. The graph in Fig. 12a does not satisfy Eq. (31) because, for example, the binary variable of the second module in the first section $y_{1,2} = 0$ is smaller than the binary variable of the third module of the same section $y_{1,3} = 1$; therefore, this graph is not represented in the ideal MINLP representation. The graph in Fig. 12b satisfies Eq. (31). There is no other graph that would be represented and be isomorphic with this one. This is the only represented graph of all its isomorphic graphs. This applies to all the other represented graphs, as well. All the isomorphic graphs are excluded from the representation. It follows that the representation becomes ideal (IMR) by inserting Eq. (31) to NSXMR.

As the non-ideality of the representation is decreased to 0 (ideal representation), problems of greater scale are expected to be solvable. First the case of $k = 9$, insolvable by NSXMR, was solved using IMR, and then the size of the problem was increased. The solution times are collected in Table 4, and the

Table 3
Results of NSXMR

k	Cost (US$/year)	Number of iterations	Solution time (s)	NLP (s)	MILP (s)	NLP/it. (s)	MILP/it. (s)	Non-ideality
8	4619	3	125.08	0.67	124.41	0.223	41.470	706299
9	4619	–	–	–	–	–	–	6137574

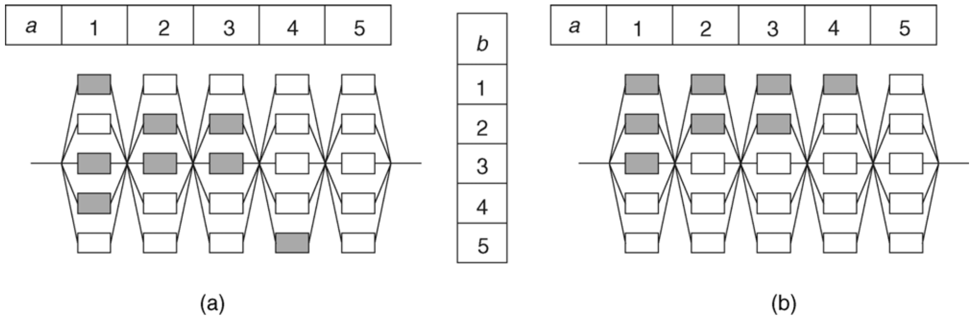


Fig. 12. (a) A non-considered graph. (b) A considered graph.

Table 4
Results of IMR

k	Cost (US$/year)	Number of iterations	Solution time (s)	NLP (s)	MILP (s)	NLP/it. (s)	MILP/it. (s)
9	4619	4	4.08	0.76	3.32	0.190	0.830
10	4619	3	2.65	0.85	1.80	0.283	0.600
11	–	–	–	–	–	–	–

solutions are shown in Figs. 13 and 14. The non-idealities are not indicated in Table 4 because the representation is ideal, and thus the non-ideality is 0.

Solution was found in case of $k = 9$ and 10; the 11×11 problem was found insolvable with IMR.

The solution time of the NLP subproblems slightly increased because the number of equations and variables were increased with the problem size. On the other hand, the search space and the solution time of the MILP subproblem were decreased drastically, by two orders of magnitude, as a result of excluding the isomorphic graphs from the representation.

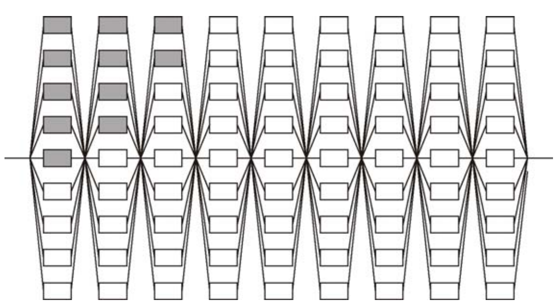


Fig. 13. Optimal solution with IMR, $k = 9$.

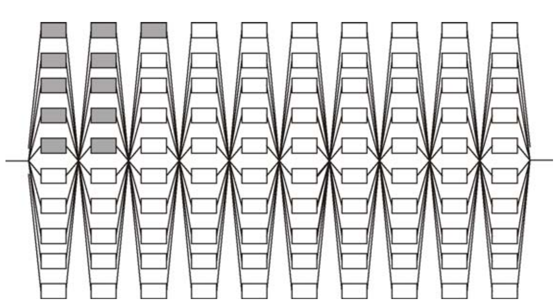


Fig. 14. Optimal solution with IMR, $k = 10$.

The maximum size of problems solvable with CMR was $7 \times 7 = 49$. The solvable size was increased by one, to $8 \times 8 = 64$, by excluding the graphs of non-considered structures (using NSXMR). And finally, even the $10 \times 10 = 100$ problem became solvable, and solution time was decreased drastically, as a result of excluding the isomorphic graphs and using ideal representation. No further constraints can be added to the representation because the representation is already ideal. Thus, the size of the solvable problems cannot be increased further this way.

5.4. *Decreased number of binary variables*

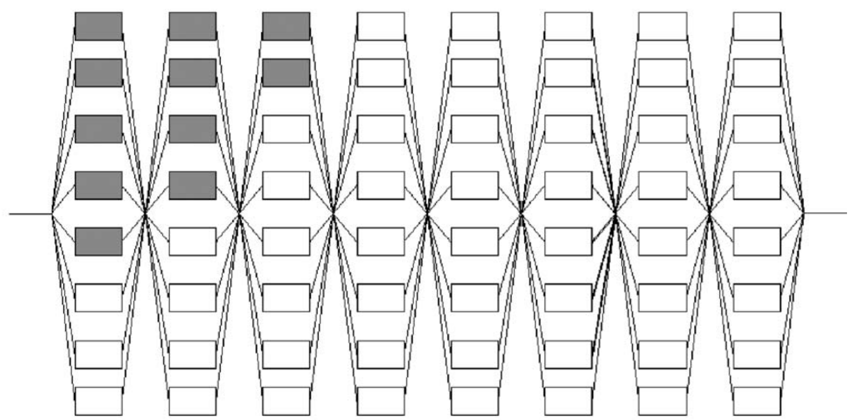
Further advance is expected by decreasing the number of binary variables. The minimum number of binary variables is suggested to use. However, this minimum number cannot be calculated in this case because the number of represented graphs is not known exactly. We have only an approximation for this minimum. But any kind of decrease in the number of binary variables is expected to enhance the solution.

A possible way to decrease the number of binary variables is using minimum number of binary variables in each section. The number of membrane modules in a section can be expressed as the sum of the binary variables in the section (Eq. (32)) if either CMR, NSXMR or IMR is applied:

$$\mathrm{UNITS}_a = \sum_{b=1}^{k} y_{a,b}, \quad a = 1, 2, \ldots, k \tag{32}$$

In order to use minimal number of binary variables in section a, the number of membrane modules is expressed in binary number system, where the binary variable $\hat{y}_{a,x}$ denotes the binary digit belonging to 2^{x-1}:

$$\mathrm{UNITS}_a = \hat{y}_{a,1} + 2 \cdot \hat{y}_{a,2} + 4 \cdot \hat{y}_{a,3} + 8 \cdot \hat{y}_{a,4} \tag{33}$$



a	1	2	3	4	5	6	7	8
$\hat{y}_{a,1}$	1	0	0	0	0	0	0	0
$\hat{y}_{a,2}$	0	0	1	0	0	0	0	0
$\hat{y}_{a,3}$	1	1	0	0	0	0	0	0
$\hat{y}_{a,4}$	0	0	0	0	0	0	0	0
$UNITS_a$	5	4	2	0	0	0	0	0

Fig. 15. A graph showing the use of Eq. (33).

The use of Eq. (33) is shown in Fig. 15. There are five membrane modules in the first section. This can be expressed with the help of new binary variables using Eq. (34). The number of membrane modules in the second and third sections can be calculated similarly.

$$\mathrm{UNITS}_1 = \hat{y}_{1,1} + 2\cdot\hat{y}_{1,2} + 4\cdot\hat{y}_{1,3} + 8\cdot\hat{y}_{1,4} = 1\cdot 1 + 2\cdot 0 + 4\cdot 1 + 8\cdot 0 = 5 \quad (34)$$

Each variable UNITS_a can take integer value from 0 to 15, as a result of Eq. (33). But maximum k modules can be built in a section; therefore, an upper bound is given to UNITS_a:

$$\mathrm{UNITS}_a \leq k \quad (35)$$

The graphs of non-considered structures were earlier excluded from the representation by Eq. (30). This criterion has to be reformulated using the new binary variables:

$$\mathrm{UNITS}_a \geq \mathrm{UNITS}_{a+1}, \quad a = 1, 2, \ldots, k-1 \quad (36)$$

Eq. (31) applied to exclude the isomorphic graphs from representation when using IMR, is not necessary when the new binary variables are used because only the number of membrane modules in a section is determined by these variables, and not the actual placement of the modules themselves. Therefore, isomorphic graphs cannot be represented, and Eq. (31) is omitted.

By using Eqs. (33), (35) and (36), the representation becomes ideal because only considered graphs are represented. This representation is thus ideal and is characterized with a decreased number of binary variables. Therefore, this is referred to as 'binarily decreased ideal MR' (BDIMR).

In case of $k = 11$, 121 binary variables were used in IMR, but the optimum could not be found. Using BDIMR, other kind of binary variables are assigned to all sections; therefore, $4 \times 11 = 44$ binary variables are used. Thus, the number of binary variables is decreased almost to its fourth.

Approximately 3.99E+8 *considered* graphs are potentially present in a 16×16 membrane system. For constructing binarily minimal representation, $\log_2(3.99\mathrm{E}{+}8) = 28.58 \leq 29$ binary variables ought to be used. Instead, $k \times 5 = 90$ binary variables are used in BDIMR. The minimal number of binary variables is computed using an estimate from below; therefore, it is assumed that our BDIMR is not a binarily minimal representation.

BDIMR was first applied to a $k = 11$ membrane system. The size of the problem was then increased. In case of $k = 16$, Eq. (33) is extended so that UNITS_a can take the value above 15 (Eq. (37)):

$$\mathrm{UNITS}_a = \hat{y}_{a,1} + 2\cdot\hat{y}_{a,2} + 4\cdot\hat{y}_{a,3} + 8\cdot\hat{y}_{a,4} + 16\cdot\hat{y}_{a,5} \quad (37)$$

The results are shown in Figs. 16 and 17, the solution times are collected in Table 5. From $k = 17$ an on, the problem is insolvable.

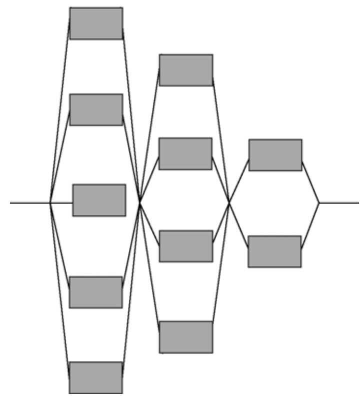


Fig. 16. Optimal solution with BDIMR, $k = 11, 12, 13, 16$.

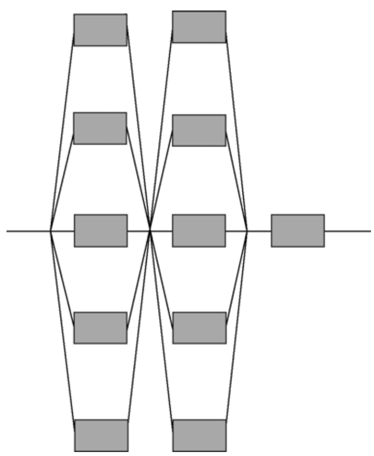


Fig. 17. Optimal solution with BDIMR, $k = 14, 15$.

The NLP solution time increases with the problem size. In case of $k = 14$ and 15, the NLP solution time per iteration is a bit greater than expected because of the smaller number of iterations. The MILP solution time increases almost linearly with a very small slope, except in the cases $k = 14$ and 15, where MILP time decreased almost to its third.

5.5. Comparison of the representations

Four MINLP representations were applied to the same pervaporation system: a conventional representation (CMR) that is similar to the basic representation BMR, but the redundant and unnecessary equations were already excluded, a representation excluding the non-considered structures (NSXMR), an ideal one (IMR), and an ideal and binarily decreased representation (BDIMR). Each representation is an improvement of the preceding one.

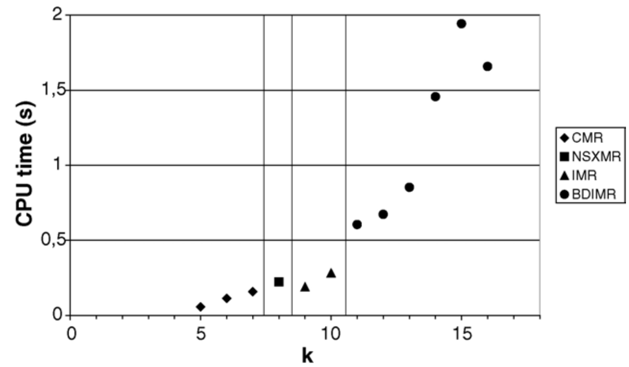


Fig. 18. NLP solution times per iterations.

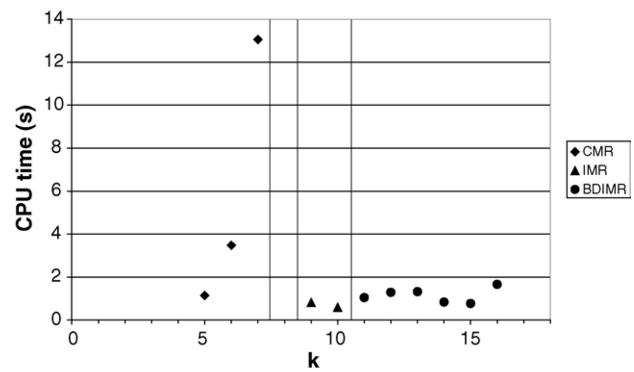


Fig. 19. MILP solution times per iterations.

The solution time data and results of all the MINLP representations are summarized in Table 5. With the exception of case $k = 5$, the number of iterations is either 3 or 4 in all the cases.

The NLP solution time per iterations (Fig. 18) increases exponentially with the problem size. This is caused most probably by the increasing number of equations. The number of binary variables does not have any effect on the NLP solution time because the binary values are fixed in the NLP subproblems.

The MILP solution time per iterations increases exponentially with the problem size in case of CMR and NSXMR.

Table 5
Results of BDIMR

k	Cost (US$/year)	Number of iterations	Solution time (s)	NLP (s)	MILP (s)	NLP/it. (s)	MILP/it. (s)
11	4619	4	6.6	2.42	4.18	0.605	1.045
12	4619	4	7.84	2.69	5.15	0.673	1.288
13	4619	4	8.71	3.41	5.30	0.853	1.325
14	4619	3	6.88	4.37	2.51	1.457	0.837
15	4619	3	8.15	5.83	2.32	1.943	0.773
16	4619	4	13.28	6.63	6.65	1.658	1.663
17	–	–	–	–	–	–	–

Table 6
Summary of the computation results for Example 2

MINLP representation	k	Cost (US$/year)	Number of iterations	Solution time (s)	NLP (s)	MILP (s)	NLP/it. (s)	MILP/it. (s)	Non-ideality
CMR	5	4619	7	8.42	0.39	8.03	0.056	1.147	11515
	6	4619	3	10.77	0.34	10.43	0.113	3.477	207490
	7	4619	3	39.65	0.47	39.18	0.157	13.060	3784425
	8	–	–	–	–	–	–	–	69023979
NSXMR	8	4619	3	125.08	0.67	124.41	0.223	41.470	706299
	9	4619	–	–	–	–	–	–	6137574
IMR	9	4619	4	4.08	0.76	3.32	0.190	0.830	0
	10	4619	3	2.65	0.85	1.80	0.283	0.600	0
	11	–	–	–	–	–	–	–	0
BDIMR	11	4619	4	6.6	2.42	4.18	0.605	1.045	0
	12	4619	4	7.84	2.69	5.15	0.673	1.288	0
	13	4619	4	8.71	3.41	5.30	0.853	1.325	0
	14	4619	3	6.88	4.37	2.51	1.457	0.837	0
	15	4619	3	8.15	5.83	2.32	1.943	0.773	0
	16	4619	4	13.28	6.63	6.65	1.658	1.663	0
	17	–	–	–	–	–	–	–	0

Data for CMR are shown, but the point of NSXMR is not represented in Fig. 19 because of its high value. The reason of this exponential increase lies in the exponential increase in the number of non-considered but represented isomorphic graphs. The isomorphic graphs are excluded from the representation using IMR and BDIMR, and the increase of MILP solution time became linear. The solution time with BDIMR at $k = 16$ (including maximum 256 membrane modules) is comparable with the solution time with CMR at $k = 5$ (including maximum 25 membrane modules).

6. Conclusion

Based on the results of Part I, ideal MINLP representation is defined. Its practical use is representing all the considered graphs and no others. Non-ideality of MR is defined as the ratio of the number of represented graphs to the number of considered ones, minus 1. This measure equals zero for IMR, and reaches very high value if the number of isomorphic graphs is great. IMR can always be constructed by first expressing in conjunctive normal form the logical constraints of considering a graph, and then transforming this conjunctive normal form into linear equations with binary variables.

Binarily minimal MINLP representation is defined as MR using minimum number of binary variables with the constraints that the structural variants are all distinguished by binary variables and that the binary members in the equations are linear. The minimum number of binary variables is the ceiling of the binary logarithm of the number of considered graphs. BMMR can always be constructed, and a way of such construction is presented. This particular construction utilizes the disjunctive normal form.

The simple and small synthesis problem of Kocis and Grossmann (1987), that was used for demonstrating the construction of BGR, BMR and MR in Part I, has been applied for comparing the properties of different MINLP representations from the viewpoint of solvability. (1) The original MR of Kocis, (2) the BMR, (3) the IMR and (4) the binarily minimal and simultaneously ideal BMIMR of the same synthesis problem are tested on GAMS DICOPT++. The number of necessary iterations is decreased to its third, and the solution time is decreased to 39%, even for this rather small example, as a consequence of idealizing the representation and minimizing the number of binary variables.

Different MINLP representations of an industrial scale synthesis problem presented by Szitkai et al. (2002) have been compared. A series of maximum k membrane sections each containing maximum k number of parallelly applied membrane units of the same scale are given as a superstructure. This problem involves a high value of structural multiplicity.

(1) The conventional representation CMR, (2) a representation with excluding the graphs of non-considered structures NSXMR, (3) the ideal representation IMR and (4) the IMR with decreased (although not minimal) number of binary variables, called BDIMR are compared with solving their application to different scale (k) superstructures with GAMS DICOPT++ with CONOPT2 NLP-solver and OSL MINLP-solver. The problem scale is quadratic in k. The results are compared in Table 6.

The maximum k of the solvable problems is 7 with CMR, 8 with NSXMR, 10 with IMR and 16 with BDMIR. The solution time of BDIMR for $k = 16$ is smaller than the solution time of CMR for $k = 7$. Both idealization of the representation and reduction of the number of binary variables enhance the solvability and decreases the solution time in a great extent.

Acknowledgement

This study was partially supported by OTKA T037191 and OTKA F046282.

Appendix A. The basic MINLP representation of Example 1 in Part I

$$\begin{aligned}&-2.04(1-y_2)+\varepsilon \le x_{2,\text{in}} \le 2.04y_2\\&-1.53(1-y_3)+\varepsilon \le x_{3,\text{in}} \le 1.53y_3\end{aligned} \tag{32}$$

$$\begin{aligned}&-3.57(1-y_1)+\varepsilon \le x_{1,\text{out}} \le 3.57y_1\\&-1.11(1-y_2)+\varepsilon \le x_{2,\text{out}} \le 1.11y_2\\&-1.11(1-y_3)+\varepsilon \le x_{3,\text{out}} \le 1.11y_3\\&-1.11(1-y_4)+\varepsilon \le x_{4,\text{out}} \le 1.11y_4\end{aligned} \tag{33}$$

$$\begin{aligned}&0 \le c_{\text{fix},1} \le 0y_1\\&0 \le c_{\text{fix},2} \le 1y_2\\&0 \le c_{\text{fix},3} \le 1.5y_3\\&0 \le c_{\text{fix},4} \le 0y_4\end{aligned} \tag{34}$$

$$\begin{aligned}&0y_1 \le c_{\text{var},1} \le 6.42y_1\\&0y_2 \le c_{\text{var},2} \le 5y_2\\&0y_3 \le c_{\text{var},3} \le 2.31y_3\\&0y_4 \le c_{\text{var},4} \le 7.78y_4\end{aligned} \tag{35}$$

$$\begin{aligned}&-1.11y_2 \le x_{2,\text{out}} - \ln(x_{2,\text{in}}+1) \le 1.11(1-y_2)\\&-1.11y_2 \le -x_{2,\text{out}} + \ln(x_{2,\text{in}}+1) \le 1.11(1-y_2)\\&-1.11y_3 \le x_{3,\text{out}} - 1.2\ln(x_{3,\text{in}}+1) \le 1.11(1-y_3)\\&-1.11y_3 \le -x_{3,\text{out}} + 1.2\ln(x_{3,\text{in}}+1) \le 1.11(1-y_3)\end{aligned} \tag{36}$$

$$\begin{aligned}&0(1-y_1) \le c_{\text{fix},1} - 0 \le 0(1-y_1)\\&-1(1-y_2) \le c_{\text{fix},2} - 1 \le 0(1-y_2)\\&-1.5(1-y_3) \le c_{\text{fix},3} - 1.5 \le 0(1-y_3)\\&0(1-y_4) \le c_{\text{fix},4} - 0 \le 0(1-y_4)\end{aligned} \tag{37}$$

$$\begin{aligned}&-6.42(1-y_1) \le c_{\text{var},1} - 1.8x_{1,\text{out}} \le 6.42(1-y_1)\\&-5(1-y_2) \le c_{\text{var},2} - x_{2,\text{out}} \le 5(1-y_2)\\&-2.31(1-y_3) \le c_{\text{var},3} - 1.2x_{3,\text{out}} \le 2.31(1-y_3)\\&-7.78(1-y_4) \le c_{\text{var},4} - 7x_{4,\text{out}} \le 7.78(1-y_4)\end{aligned} \tag{38}$$

References

Brendel, M. H., Friedler, F., & Fan, L. T. (2000). Combinatorial foundation for logical formulation in process network synthesis. *Computers & Chemical Engineering*, *24*(8), 1859–1864.

Brooke, A., Kendrick, D., & Meeraus, A. (1988). *GAMS: A user's guide*. Palo Alto, CA: Scientific Press.

Farkas, T. (2001). Optimal MINLP representation variants of superstructures. In *Study report at conference on scientific research by students (TDK)* (in Hungarian).

Farkas, T., Rev, E., & Lelkes, Z. (2005). Process flowsheet superstructures: Structural multiplicity and redundancy. Part I: Basic GDP and MINLP representations. *Computers and Chemical Engineering*, *29*(10), 2180–2197.

Grossmann, I. E. (1996). Mixed-integer optimization techniques for algorithmic process synthesis. *Advances in Chemical Engineering*, *23*, 171.

Kocis, G. R., & Grossmann, I. E. (1987). Global optimization of nonconvex MINLP problems in process synthesis. *Industrial & Engineering Chemical Research*, *27*, 1407.

Raman, R., & Grossmann, I. E. (1991). Relation between MILP modelling and logical inference for chemical process synthesis. *Computers & Chemical Engineering*, *15*(2), 73–84.

Rev, E., Farkas, T., & Lelkes, Z. Process flowsheet structures. *International Journal of Computer Mathematics*, submitted for publication.

Szitkai, Z., Lelkes, Z., Rev, E., & Fonyo, Z. (2002). Handling of removable discontinuities in MINLP models for process synthesis problems, formulations of the Kremser equation. *Computers & Chemical Engineering*, *26*(11), 1501–1516.