Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computers and Chemical Engineering 29 (2005) 2180–2197

Computers & Chemical Engineering

www.elsevier.com/locate/compchemeng

Process flowsheet superstructures: Structural multiplicity and redundancy Part I: Basic GDP and MINLP representations

Tivadar Farkas, Endre Rev, Zoltan Lelkes *

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

Received 9 September 2004; received in revised form 11 July 2005; accepted 25 July 2005
Available online 6 September 2005

Abstract

Structural multiplicity has a significant effect on the solution of an MINLP model for process synthesis problems. The optimization model may also have built-in redundancy that cannot always be directly derived from the multiplicity of the superstructure.

A basic GDP representation (BGR) involving logical relations is defined, and can be constructed by applying a standard natural representation of the process. Basic MINLP representation (BMR) is defined by transforming the logical relations to algebraic ones.

MINLP representation (MR) is defined 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 defined. If the supergraph, i.e. the graph representing the superstructure, is structurally redundant (i.e. there are isomorphic graphs amongst their subgraphs) then BMR has binary multiplicity. Conversely, the structural redundancy of the graph does not follow from the binary multiplicity of its BMR.

Different kinds of multiplicity and redundancy measures of the MINLP representation will be defined in Part II of this series in order to help inventing tools for decreasing their detrimental effect. Alternative MINLP representations will there be defined, constructed, and compared from the viewpoint of ideality, minimality, and solution properties.

PACS: 02.60.Pn; 02.70.Rw

Keywords: Multiplicity; Superstructure; Redundancy; MINLP; Representation; Process synthesis

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.

Superstructures are to be constructed in a way to incorporate all the considered feasible process structures. On the other hand, superstructures ought to be minimal in the sense that they should include the minimum number of non-considered structures, for technical reasons. This is not a problem of mathematics. Once all the considered structures are either explicitly known or implicitly given, a superstruc-

DOI of original article:10.1016/j.compchemeng.2005.07.008.

Abbreviations: BGR, basic GDP representation; BMIMR, binarily minimal and ideal MINLP representation; BMR, basic MINLP representation; BMMR, binarily minimal MINLP representation; GDP, generalized disjunctive programming; IMR, ideal MINLP representation; MINLP, mixed integer non-linear programming; MR, MINLP representation; NLP, non-linear programming; SEN, state–equipment network; STN, state–task network

* Corresponding author. Fax: +36 1 463 3197.
E-mail address: lelkes@mail.bme.hu (Z. Lelkes).

0098-1354/$ – see front matter
doi:10.1016/j.compchemeng.2005.07.007

Nomenclature

Parameters

L	lower bound
U	upper bound
ε	small positive value

Variables

a	stream in Example 1
b	stream in Example 1
c	stream in Example 1
c	cost
d	design/control variable
e	extensive variable
i	intensive variable
o	operation variable
x	continuous variable
y	binary variable
z	logical variable
φ	fraction variable, belonging to edge

Sets and regions

B	subset of feasible region
E	set of edges
FR	feasible region
M	set of units
M_{cond}	set of conditional units
M_{perm}	set of permanent units
$\mathbf{X}$	region of continuous variables
$\mathbf{Y}$	region of binary variables
$\mathbf{Z}$	region of logical variables
α	number of input ports
β	number of output ports

Subscripts

e	edge
fix	fix
in	inlet
k	port
h	port
m	unit
n	unit
out	outlet
t	type of unit
var	variable

Superscripts

max	maximum
UP	upper bound

Functions

P	function describing the unit operation and its equipment
P_{fix}	fix cost function of a unit
P_{var}	variable cost function of a unit
Z	objective function

ture can be constructed using mathematics (Friedler, Fan, & Imreh, 1998; Friedler, Tarjan, Huang, & Fan, 1992, 1993). The real problem, however, is of engineering nature. Delimiting the set of all the considered structures is a rather difficult, if not impossible, chemical engineering task related to the engineer's insight to the whole complexity of technical, economical, and human circumstances.

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. Although some researchers apply intuitive methodology for decreasing the multiplicity, it is worth to analyze 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. Yeomans and Grossmann (1999) were the first who developed a general method for constructing MINLP formulation for two kinds of superstructures, namely state–task networks (STN) and state–equipment networks (SEN). Generalized disjunctive programming (GDP) is used for representing a superstructure. They also developed (Lee & Grossmann, 2000) a special algorithm and computer program for this type of MINLP representation.

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 here is to develop a methodology for constructing MINLP representations of superstructures most appropriate for numerical solution. Therefore, in the present article a definition of the MINLP representation conform with the R-graph representation is developed.

Different kinds of multiplicity and redundancy measures of the MINLP representation will be defined in Part II of this series to help inventing tools for decreasing their detrimental effect, and alternative MINLP representations will in that part be examined and compared from the viewpoint of ideality, minimality, and solution properties.

2. Representation of structures by graphs

Process synthesis aims at inventing the optimal construction of a targeted process, together with the optimal operation policy, by optimally selecting from the available operation units and assigning their connection. Selection of units together with the connection is called process structure.

Usually infinite number of physically feasible process structures can be used to achieve the targeted chemical process, although most of them are far from being optimal. For applying MINLP technique with superstructure approach, the engineer have to select from the physically feasible structures a finite set of process structures to be studied. These preliminary selected process structures will here be called *considered structures*. This set of considered structures can be asssigned by any explicite or implicite way.

Once this set of considered structures is given, the engineer should be able to construct a mathematically treatable superstructure that includes all the considered structures as its substructures. This superstructure itself is usually represented by a graph or network and/or by a system of variables and mathematical relations. The graph representation of the superstructure will here be called *the supergraph*. Unfortunately, the same considered structure may be represented by different subgraphs of the same supergraph (see Rev et al., submitted for publication). The supergraph of a proper superstructure should contain at least one subgraph to each of the considered process structures to be represented.

However, the superstructure may also contain substructures that are not considered. A superstructure s of structures $s_1, s_2, \ldots$ is called ideal if each substructure of s is also a substructure of one of those structures $s_1, s_2, \ldots$ (Definition 12 in Rev et al., submitted for publication). It is shown by Theorem 2 in Rev et al. (submitted for publication) that there is no ideal superstructure of two structures if neither of them is a substructure of the other one. As usually this is the case, the superstructure is expected to contain structures that are not considered.

A supersturcture may also contain structures that are not even physically feasible; these are *infeasible structures*. Those feasible structures that are not considered, together with the physically infeasible structures are simply called non-considered structures. The system of subsets in the set of substructures of the superstructure is illustrated in Fig. 1.

2.1. R-graphs

The treatment of structures are greatly enhanced if they are, as it usually happens, represented by networks or graphs.

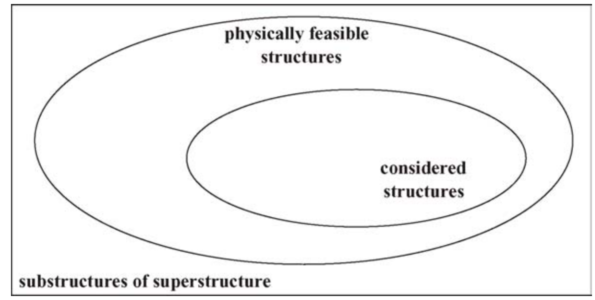

Fig. 1. Classification of structures.

Therefore, the relation between graphs and structures have to be clarified before discussing their characteristics. In order to have exact definition, we invented a special kind of graph, the so-called *R-graph* (Rev et al., submitted for publication). The R-graph is a one task–one equipment graph. The nodes of an R-graph are not units but the input and output ports of the possible units. The directed edges correspond to streams, which always start from an output port node of a unit and end on an input port node of a unit. The output port nodes are treated as arbitrary stream splitters, whereas the input nodes as arbitrary unifiers. An example R-graph is shown in Fig. 2.

An R-graph is a graph also in mathematical sense. It means, that all the edges start from a node and end on a node. Therefore, source and sink units are included in order to prevent edges to start from or end on outside the graph. The source units (e.g. Unit 1 in Fig. 2) have no input nodes, and the sink units (e.g. Unit 4 in Fig. 2) have no output nodes.

The subgraph of an R-graph is a short-hand for R-subgraph of an R-graph, i.e. it is also an R-graph. All the nodes have to be connected in a subgraph, i.e. in an R-subgraph. A counter-example is shown in Fig. 3, where the second input node of Unit 2 is not connected to any other node. This is not an R-graph even if the engineer can easily assign meaning to this figure by considering the unoccupied port as a practically non-existing one. If that kind of unit may work with just one feed then a unit type with one input port should also be listed in the frame set of units, and used in the graph. If both solutions are possible then both types should be applied, e.g. in parallel connection, as is shown in Fig. 4. Here bb is a

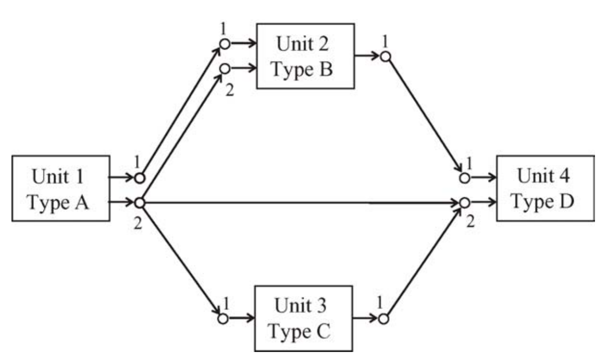

Fig. 2. An example R-graph.

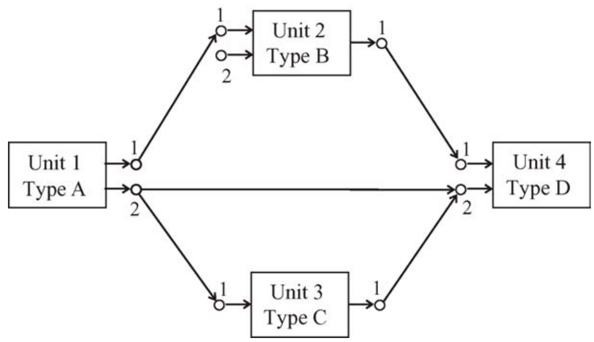

Fig. 3. A counter-example. This is not an R-graph because there is an unconnected node.

unit type performing almost the same task, from engineering viewpoint, as type B, but with one feed only.

One of the main advantages of using R-graph is that in this way by-pass redundancy is avoided. About by-pass redundancy see more in Appendix A and Rev et al. (submitted for publication).

In the present article, from here on, we will apply the simple term graph instead of R-graph.

2.2. Redundancy and structural multiplicity

Structural multiplicity is an important phenomenon caused by the possibility of representing the same structure with different graphs. See, as an over-simplified example, Fig. 5a and b. A graph of the superstructure is shown in Fig. 5a. This graph has several subgraphs, representing substructures of the superstructure, two of which are shown in Fig. 5b. The types of Units 2 and 3 are identical (Type B); they are two copies of the same unit type. Therefore, these subgraphs represent the same process structure, but they are different because the nodes and edges of the supergraphs are unambiguously denominated. (Graphs, by mathematical sense, are constructed from labeled entities.) These two R-graphs are called isomorphic because they are identical neglecting the differences in different copies of units of the same type. In other words they are isomorphic because one of them can be transformed to the other one merely by re-naming the units. Consequently, the *structures are* not equivalent to graphs but to *sets of isomorphic* (or, in other word: equivalent) *graphs* (see Rev et al., submitted for publication). They can be represented by any of these isomorphic graphs. The isomorphic graphs are not necessarily represented distinctly when a mathematical model is constructed, because they correspond to the same process structure.

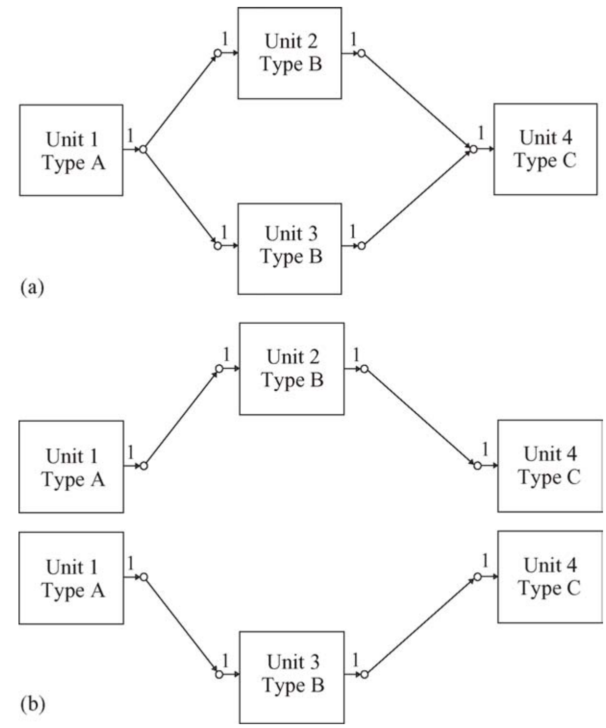

Fig. 5. (a) Graph of a superstructure. (b) Isomorphic subgraphs representing the same structure.

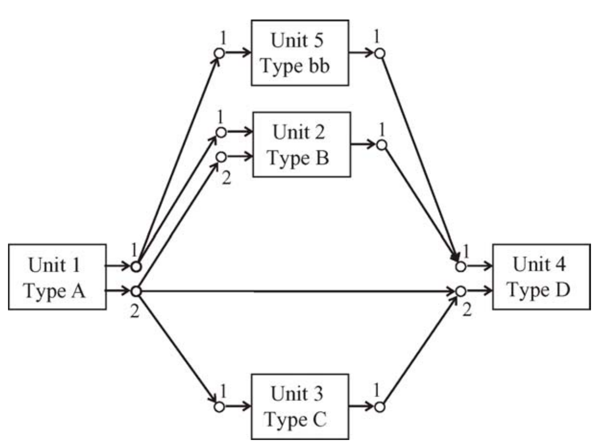

Fig. 4. An R-graph involving a unit of B type with two feeds and a unit of bb type with one feed.

Structural multiplicity of a structure s in a superstructure s^* represented by a graph r^* is defined as the number of all those subgraphs of r^* that represent s. In the example above, the multiplicity of the structure represented by the graphs shown in Fig. 5b is 2 inside the superstructure represented by the graph shown in Fig. 5a.

On the other hand, redundancy of a structure in itself is defined as the difference between the number of all the subgraphs of a graph representing the structure and the number of all the substructures of the studied structure.

As outlined above, the set of the substructures of a superstructure is classified according to whether the structure is physically feasible, and if yes then whether it is considered by the engineer. The same distinction can be made amongst the graphs representing the structures.

Any graph representing physically feasible structure is, naturally, called feasible graph. All the other graphs (if they are anyhow constructed) are represented by so-called infeasible graphs. The relation between considered structures and their representing graphs is just a bit more complicated

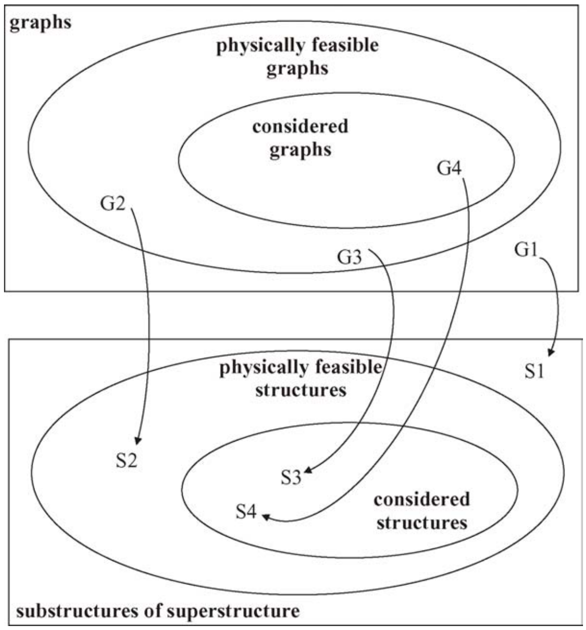

Fig. 6. Relations between feasible and considered structures and graphs.

because considered structures may be represented by considered graphs and non-considered graphs as well.

Since the detrimental effect of structural multiplicity and redundancy can be decreased by non-considering isomorphic graphs representing the same structure, we define the set of considered graphs to be such a set of graphs that each considered structure is represented by exactly one graph in this set, and each element of this set represents a considered structure. It follows, that there are no isomorphic graphs in the set of considered graphs.

The relations between the mentioned sets are shown in Fig. 6.

3. Example 1—Problem statement

The notions explicated in Section 2 are demonstrated in this small example. A small planning problem (Kocis & Grossmann, 1987) is considered to demonstrate both the new concepts and how the methodology works. The process is to produce product C from raw materials A and/or B with maximal profit. Three units can be used to accomplish this aim, as it is shown in Fig. 7. Data are given in Table 1.

Fig. 8 is a "one task–one equipment network", used for visualising the superstructure. The units are represented by nodes, and the streams are represented by edges. This figure does not represent a graph in mathematical sense because in graphs the edges must connect nodes, whereas here some edges (namely a_1 and b_1) originate from outside, and one (namely c) is directed to outside of the process. For the sake of exact discussion, the graph (R-graph) representation of the same system is shown in Fig. 9.

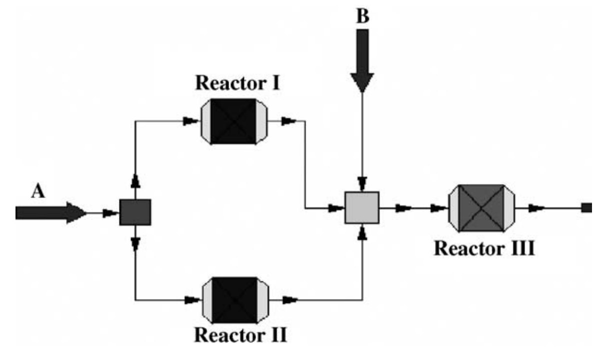

Fig. 7. Example superstructure.

Table 1
Data for Example problem 1

Unit	Fixed cost ($\times 10^3$ US\$/h)	Variable cost ($\times 10^3$ US\$/tonnes of product)
I	1.0	1.0
II	1.5	1.2
III	3.5	2.0
Raw material costs, $\times 10^3$ US\$		1.8/tonnes of A, 7.0/tonnes of B
Revenue, $\times 10^3$ US\$		13.0/tonnes of C
Mass balances for units		
Unit I: $b_2 = \ln(1 + a_2)$		
Unit II: $b_3 = 1.2 \ln(1 + a_3)$		
Unit III: $c = 0.9b$		
$c \leq 1$		
$b_2 \geq 5$		

There are three additional units here. Units 1 and 4 are the sources of raw materials A and B; Unit 6 is the sink of product C. The source units do not have input ports, and the sink unit does not have output port.

In this example *the considered structures* are selected according to the following criteria:

1. Source stream(s) should exist. (That is, either stream a_1, or b_1, or both should be present.)
2. Product stream(s) should exist. (That is, stream c should be present.)
3. No other outlet stream than the product stream(s) may be present. (That is, only c may be an outlet stream here.)
4. Units I and II (in the graph representation: Units 2 and 3) should not exist simultaneously.

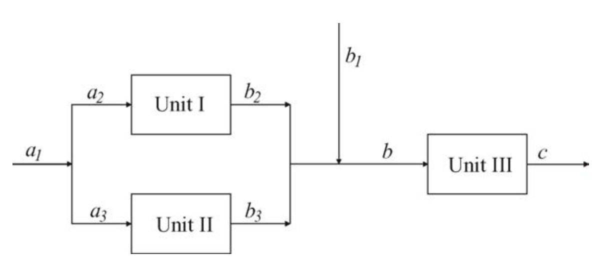

Fig. 8. Example network.

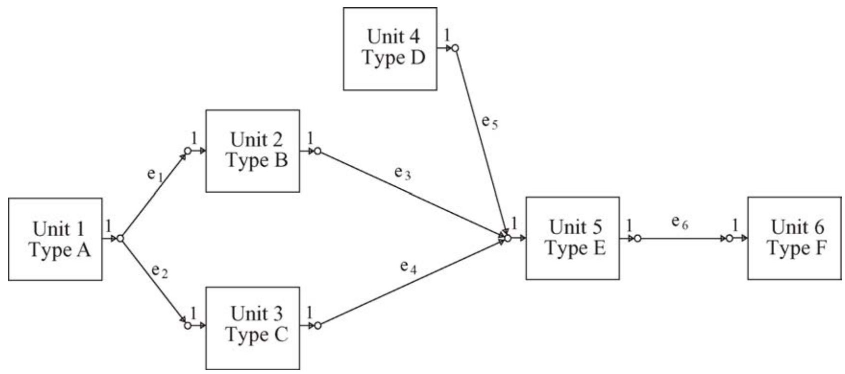

Fig. 9. Graph (R-graph) representation of example problem 1. Letter ‘e’ stands for edges.

The first criterion states that the product is always made from the raw material(s), and cannot be made from nothing. The second criterion states that some product is to be produced. The third criterion expresses the special need of this example that no side-product is to be produced. The fourth criterion is applied because Units I and II (Units 2 and 3 in the graph representation) accomplish the same (or similar) transformation on the materials, even if they are units of different type, and application of two units of different types parallelly on the same task seems uneconomic. All these criteria are based on engineering considerations.

Before presenting the graph representation of the considered structures, please, check the network representation. The network shown in Fig. 10 is a subnetwork of the network shown in Fig. 8. However, it does not satisfy the above criteria; thus, it represents a non-considered structure. The chance for such a problem to occur is significantly reduced by applying R-graphs representation.

Fig. 11a–g show all the subgraphs (R-subgraphs) of the supergraph shown in Fig. 9. All these subgraphs satisfy the first three criteria. On the other hand, Graphs 6 and 7 (Fig. 11f and g) do not satisfy the fourth criterion. The first three criteria are automatically satisfied, and only the fourth criterion is to be checked when graphs are applied. Therefore, the use of R-graph representation is beneficial in automatic synthesis.

An other benefit of the R-graph representation is that the superstructure (Fig. 7) is more similar to it (Fig. 9) than to the network representation (Fig. 8). It means that the R-graph representation is close to the modular unit approach. The main difference is the lack of stream splitters and unifiers, but it is beneficial for avoiding by-pass redundancy during the optimization.

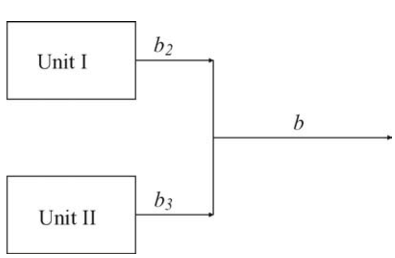

Fig. 10. A subnetwork that represents a non-considered structure.

All the units of any supergraph are different in this simple example; there are no units of the same type. For this reason, there cannot exist isomorphic subgraphs; every substructure is represented by only one graph. Therefore, each graph representing a considered structure is also a considered graph. The phenomenon of isomorphism and the corresponding redundancy and multiplicity are, therefore, not present in this example; we will encounter and treat them in Part II.

How the subgraphs of this example are sorted according to Fig. 6 is shown in Fig. 12.

4. Redundancy in representing structures by MINLP problems

The above distinction between the set of considered structures and considered graphs is crucial in representing the structures by algebraic/logical models. Such a model contains variables; those variables correspond to uniquely labeled elements of the graph actually representing the superstructure. Therefore, the MINLP problem is not directly applied to the superstructure but to one of its particular representing graphs, the so-called supergraph. A solution of the MINLP problem assigns a subgraph of that particular representing supergraph.

We have already defined (Rev et al., submitted for publication) structural multiplicity and structural redundancy using graph representation. We also defined ideal superstructure of a set of structures as such a superstructure of all these structures that does not have any other substructure. It is expedient to apply this definition to the set of considered structures.

Graphs, representing structures, will be further represented by MINLP problems. How can we be sure that an MINLP problem represents a graph? Later in this article, we will define MINLP representation, abbreviated as MR, for answering this question. We will define multiplicity and ideality of these MINLP representations, as well, in Part II.

In general, an MR, as an MINLP problem, is an algebraic construction Eqs. (1)–(7), denoted by P1, in the following or similar form (Kocis & Grossmann, 1987):

$$Z = \min[\mathbf{c}^{\mathrm{T}}\mathbf{y} + f(\mathbf{x})] \quad \text{(objective function)} \tag{1}$$

$$\text{s.t.} \quad \boldsymbol{h}(\mathbf{x}) = 0 \tag{2}$$

$$\boldsymbol{g}(\mathbf{x}) \leq 0 \tag{3}$$

$$\mathbf{Ax} = \mathbf{a} \tag{4}$$

$$\mathbf{By} + \mathbf{Cx} \leq \mathbf{d} \quad \text{(constraints)} \tag{5}$$

where the **x** continuous and **y** binary variables are elements of the sets:

$$\mathbf{x} \in \mathbf{X} = \{\mathbf{x} | \mathbf{x} \in R^n, \mathbf{x}^{\mathbf{L}} \leq \mathbf{x} \leq \mathbf{x}^{\mathbf{U}}\} \tag{6}$$

$$\mathbf{y} \in \mathbf{Y} = \{\mathbf{y} | \mathbf{y} \in \{0, 1\}^n, \mathbf{Ey} \leq \mathbf{e}\} \tag{7}$$

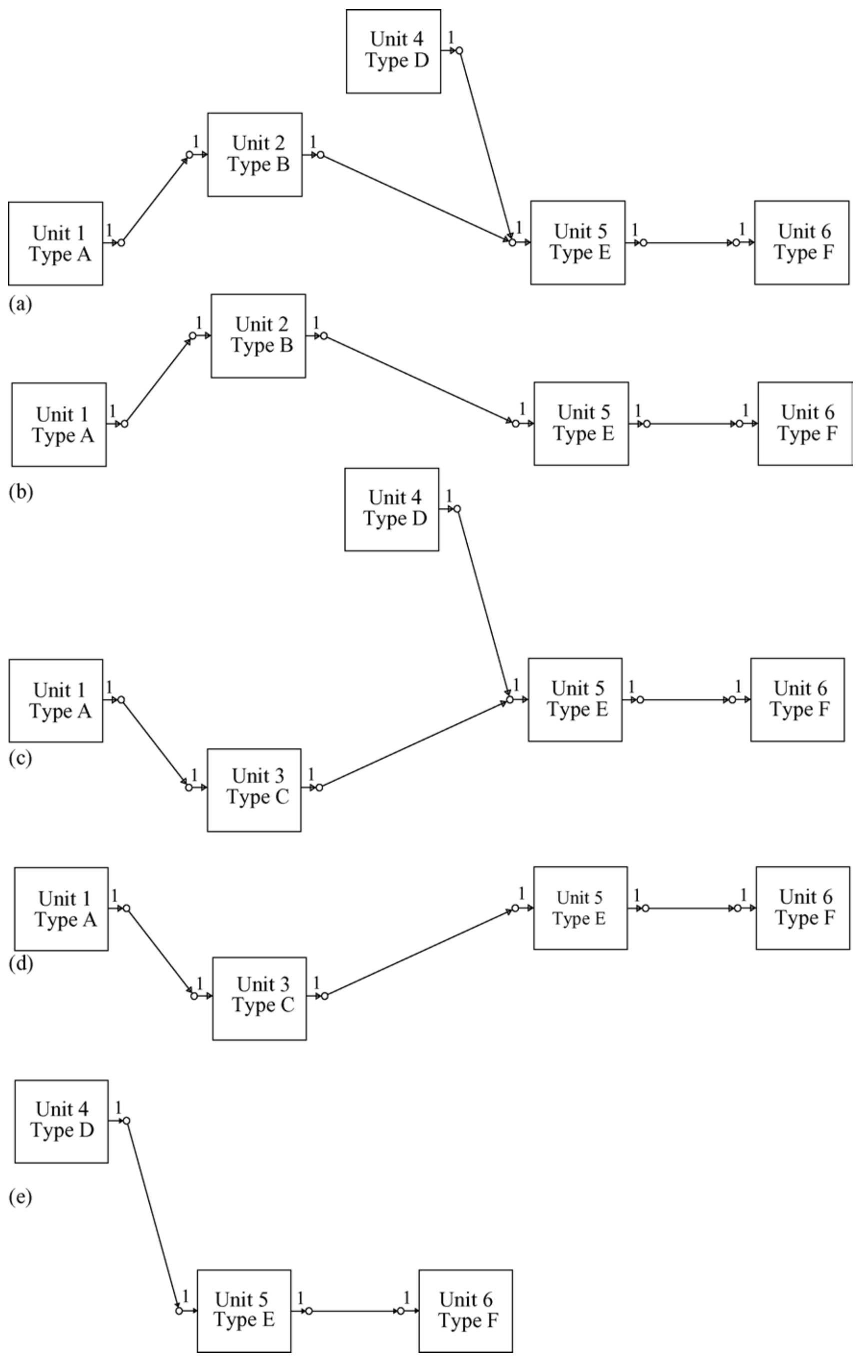

Fig. 11. (a) Graph 1: a subgraph of the supergraph. (b) Graph 2: a subgraph of the supergraph. (c) Graph 3: a subgraph of the supergraph. (d) Graph 4: a subgraph of the supergraph. (e) Graph 5: a subgraph of the supergraph. (f) Graph 6: a subgraph of the supergraph. (g) Graph 7: a subgraph of the supergraph.

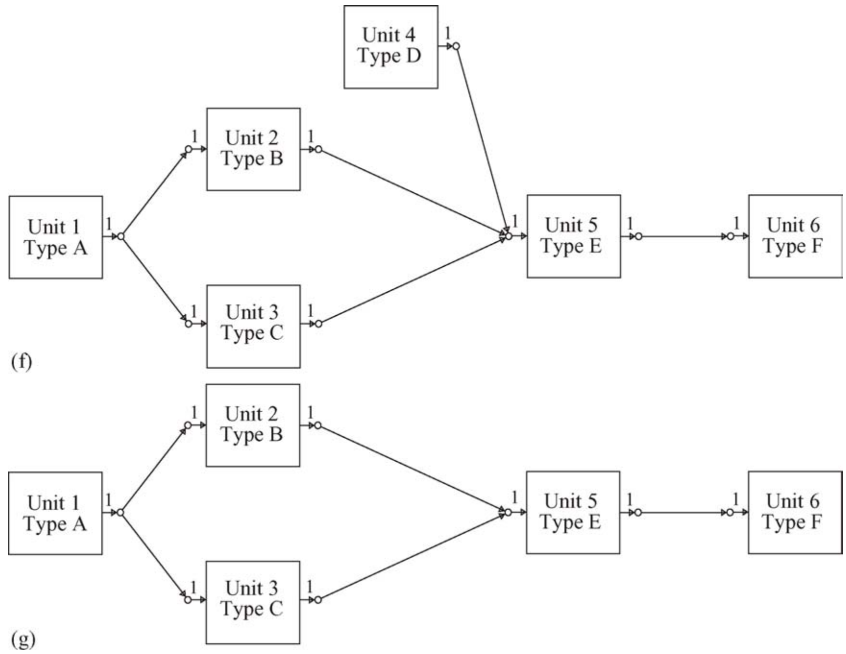

Fig. 11. (*Continued*).

and where **x** is the vector of continuous variables, typically representing flow rates, pressures, temperatures, and sizes of units, while the vector of binary variables **y** is used for representing alternative structures of the superstructure. The equations $h(\mathbf{x}) = 0$ and $\mathbf{Ax} = \mathbf{a}$ ordinarily correspond to material and energy balances and some of design equations, while equations $g(\mathbf{x}) \leq 0$ represent the process specifications. Alternative structures can be defined by the logical constraints and linear equations: $\mathbf{By} + \mathbf{Cx} \leq \mathbf{d}$ and $\mathbf{Ey} \leq \mathbf{e}$. The objective function: $Z = \min[\mathbf{c}^{\mathrm{T}}\mathbf{y} + f(\mathbf{x})]$ is often defined as a total annual cost. The term $\mathbf{c}^{\mathrm{T}}\mathbf{y}$ includes fix cost, while the function $f(\mathbf{x})$ involves operating costs, size dependent costs, and the revenues. The feasible region of P1 is denoted by

$$\mathrm{FR(P1)} = \{\mathbf{x}, \mathbf{y} | \mathbf{x} \in \mathbf{X}, \mathbf{y} \in \mathbf{Y},\ \mathbf{h}(\mathbf{x}) = 0,\ \mathbf{g}(\mathbf{x}) \leq 0, \mathbf{Ax} = \mathbf{a},\ \mathbf{By} + \mathbf{Cx} \leq \mathbf{d}\} \quad (8)$$

i.e. the set of **x** and **y** that satisfy Eqs. (2)–(5).

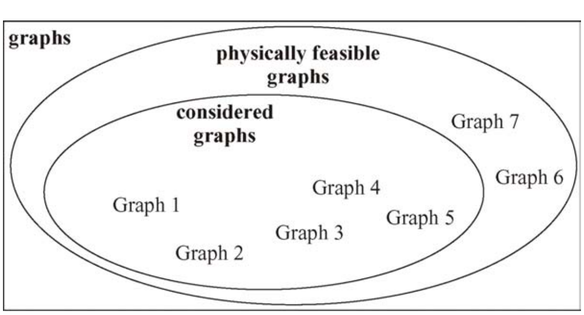

Fig. 12. Grouping of the subraphs.

The MR as usually applied to the synthesis problem is never unique. An MR as defined above is nothing to do with the synthesis problem itself unless an unambiguous mapping is given from the domain of the variable values of the MR to the domain of the value set of natural parameters of the synthesis problem, or at least to the graph representation of the superstructure. Such a mapping may be assigned by a mapping between the variables themselves, in a way similar to, for example, what is used by Yeomans and Grossmann (1999) who assigned a binary variable to each operation unit to denote its existence, and continuous variables to stream values and operational parameters.

The essential problems in mathematically defining the MR are representativeness and uniqueness. Whether an MR represents a superstructure and all its considered substructures should be determinable. An other crucial problem is comparison of two different MINLP representations to decide if some process flowsheets are represented by both of them. This seems a rather difficult task because infinite number of different MR-s can be generated just by simple variable transformations in such a way that the generated MR-s are equivalent.

In order to avoid ambiguousity in defining the variables of the MR, we first apply a basic GDP representation (BGR) involving logical relations, and then construct a so-called basic MINLP representation (BMR) with standard mapping between them. BGR can be constructed by applying a standard 'natural' representation of the process. Then, it is easy to unambiguously transform the logical relations to algebraic ones. Equivalency and representativeness of MR-s in general form will then be analyzed by reducing them to their BMR or by comparing their feasible domains.

Once the definition of MR is properly given, we are able to analyze ideality and redundancy. We will define and compare ideal MR (IMR) and binarily minimal MR (BMMR) as most important extremities, in Part II of this article series.

5. Basic representations

The usual practice of developing an MINLP model of a synthesis problem is first intuitively constructing a superstructure of the given problem. The superstructure is usually represented by a network or, more conveniently, a graph (a supergraph). Then some state variables are also intuitively assigned to the elements (nodes and edges) of the graph. These variables may refer to thermodynamic states, operational parameters, construction parameters, and even to the existence of units and/or connections. These variables together with the evaluation variables (parameters of the objective function, and the variable carrying the function value itself) constitute the set of variables of the MINLP model.

Constructing a proper MR is a difficult task demanding heavily on the engineer's abilities. Since logical relations are more human friendly than integer variables and relations, the logical formulation, e.g. the GDP representation (see Yeomans & Grossmann, 1999), can be utilized as a first step in formulating the MR. Thus, we first construct a so-called basic GDP representation (BGR) then, in turn, a basic MINLP representation (BMR) automatically transformed from BGR. This BMR will serve as a reference for comparison to deciding if an arbitrary MINLP formulation represent the superstructure.

Representativeness and uniqueness are the two essential viewpoints in defining BGR and BMR. BMR is formed in a way as to represent only and all the subgraphs of a given supergraph. Numerical viewpoints can be taken into account later, in constructing the actual form of MR. The variables and the equations defined in BGR and BMR are not necessarily present in the final form of MR.

5.1. Basic GDP representation

A general disjunctive programming (GDP) model is constructed in the spirit of Yeomans and Grossmann (1999) but based on the R-graph representation given in Rev et al. (submitted for publication). R-graph corresponds to a state equipment network with 'one task–one equipment (OTOE)' assignment in the sense of classification by Yeomans and Grossmann (1999). This GDP model is called here basic GDP representation (BGR).

It is formulated in such a way that some formulas describing the behaviour of unit operations work on the variables belonging to particular units; some formulas work on stream properties, etc. Therefore the variables have to be grouped according to which unit operations and which streams they belong to.

A set of unit type variables and unit type equations are to be attributed to each unit type. When a particular unit (with particular labels, like "1" or "2" amongst the same type of units) is selected, it is attributed with particular unit equations and constraints (together called unit relations) according to its type, acting on the set of *unit variables*. These unit relations are of the same shape for identical unit types. That is, if there are two copies of the same unit type then two, formally equivalent, sets of relations are applied to different variables.

Basic GDP representation (BGR) is then defined by the following formulation.

5.1.1. Sets

The sets include the frame set of units and unit types, their input and output ports as well. Any particular graph is a construction including the set M of actual units, the set of input and output ports, the types of these units, and the set E of graph edges e ($e \in E$).

R-graph is a connected graph, and its subgraphs also have this property. In some cases one or more units of a graph occur in each of its subgraphs. For example, if a graph has exactly one sink unit, then this unit should occur in all its subgraphs, otherwise those subgraphs would not be R-graphs. Generally not just the sink or source units, but a unit of any type may have this property in a particular supergraph. These units are called *permanent units* of the supergraph, while all the other units are called *conditional units*.

Accordingly, the set M of units is partitioned in BGR as $M = M_{\text{perm}} \cup M_{\text{cond}}$, where M_{perm} is the set of units that occur in all the subgraphs, and M_{cond} is the set of units that does not occur in all the subgraphs of the supergraph.

5.1.2. Variables

Each unit $m \in M$ is attributed with the following variables:

Numerical variables:

$\mathbf{e}_{m,\text{in},k}$	array of inlet extensive variables, $k = 1, 2, \ldots, \alpha_\text{t}$
$\mathbf{e}_{m,\text{out},k}$	array of outlet extensive variables, $k = 1, 2, \ldots, \beta_\text{t}$
$\mathbf{i}_{m,\text{in},k}$	array of inlet intensive variables, $k = 1, 2, \ldots, \alpha_\text{t}$
$\mathbf{i}_{m,\text{out},k}$	array of outlet intensive variables, $k = 1, 2, \ldots, \alpha_\text{t}$
$\mathbf{d}_m$	array of design and control variables
$\mathbf{o}_m$	array of operation variables
$c_{m,\text{fix}}$	fix costs due to unit m
$c_{m,\text{var}}$	variable costs due to unit m

where

t	type of unit m
α_t	number of input ports of unit type t
β_t	number of output ports of unit type t

The variables in the arrays $\mathbf{e}_{m,\mathrm{in},k}$ and $\mathbf{i}_{m,\mathrm{in},k}$ will be lumped together and denoted by $\mathbf{x}_{\mathrm{in}}$, and the lumped version of $\mathbf{e}_{m,\mathrm{out},k}$ and $\mathbf{i}_{m,\mathrm{out},k}$ is denoted by $\mathbf{x}_{\mathrm{out}}$.

For each conditional unit a logical variable is also defined:

z_m the existence of conditional unit m

Variables attributed to the edges of the graph, describing what fraction of the stream produced at the output port of a unit where that edge starts is directed to the input port of an other unit, where the edge points to:

$$0 \le \varphi_{\mathrm{e}} \le 1 \tag{9}$$

5.1.3. Feasibility constraints

Unit relations:

$$\left.\begin{array}{l}\mathbf{x}_{m,\mathrm{in}} \ge 0;\ \mathbf{x}_{m,\mathrm{out}} \ge 0\\ \mathbf{x}_{m,\mathrm{in}} \ne 0;\ \mathbf{x}_{m,\mathrm{out}} \ne 0\\ \mathbf{o}_m \ge 0;\ \mathbf{d}_m \ge 0\\ \mathbf{P}_{\mathrm{t}}(\mathbf{x}_{m,\mathrm{in}},\ \mathbf{x}_{m,\mathrm{out}},\mathbf{d}_m,\mathbf{o}_m) \le 0\\ c_{\mathrm{fix},m} = P_{\mathrm{fix_t}}(\mathbf{d}_m)\\ c_{\mathrm{var},m} = P_{\mathrm{var_t}}(\mathbf{x}_{m,\mathrm{in}},\mathbf{x}_{m,\mathrm{out}},\mathbf{d}_m,\mathbf{o}_m)\end{array}\right] \quad \text{for all } m \in M_{\mathrm{perm}} \tag{10a}$$

$$\left[\begin{array}{c} z_m\\ \wedge\\ \mathbf{x}_{m,\mathrm{in}} \ge 0;\ \mathbf{x}_{m,\mathrm{out}} \ge 0\\ \mathbf{x}_{m,\mathrm{in}} \ne 0;\ \mathbf{x}_{m,\mathrm{out}} \ne 0\\ \mathbf{o}_m \ge 0;\ \mathbf{d}_m \ge 0\\ \mathbf{P}_{\mathrm{t}}(\mathbf{x}_{m,\mathrm{in}},\mathbf{x}_{m,\mathrm{out}},\mathbf{d}_m,\mathbf{o}_m) \le 0\\ c_{\mathrm{fix},m} = P_{\mathrm{fix_t}}(\mathbf{d}_m)\\ c_{\mathrm{var},m} = P_{\mathrm{var_t}}(\mathbf{x}_{m,\mathrm{in}},\mathbf{x}_{m,\mathrm{out}},\mathbf{d}_m,\mathbf{o}_m)\end{array}\right] \vee \left[\begin{array}{c}\neg z_m\\ \wedge\\ \mathbf{x}_{m,\mathrm{in}} = 0;\ \mathbf{x}_{m,\mathrm{out}} = 0\\ \mathbf{o}_m = 0;\ \mathbf{d}_m = 0\\ c_{\mathrm{fix},m} = 0\\ c_{\mathrm{var},m} = 0\end{array}\right] \quad \text{for all } m \in M_{\mathrm{cond}} \tag{10b}$$

Equations for input ports:

$$\mathbf{e}_{m,\mathrm{in},k} = \sum_{e \in E[m,\mathrm{in},k]} \varphi_{\mathrm{e}} \mathbf{e}_{n,\mathrm{out},h} \quad \text{for all } \langle m,\ \mathrm{in},\ k\rangle \tag{11a}$$

$$i_{m,\mathrm{in},k} = f\left(\mathbf{x}_{\{n,\mathrm{out},h\}}\right) \quad \text{for all } \langle m,\ \mathrm{in},\ k\rangle \tag{11b}$$

where E[m,in,k] is the set of all the edges ending at input port k of unit m; these edges are started at output ports $\langle n,\mathrm{out},h\rangle$, where $\langle n,\mathrm{out},h\rangle$ is the output port h of unit n that is connected to input port k of unit m by an edge e. The array $\mathbf{x}_{\{n,\mathrm{out},h\}}$ includes all the extensive and intensive variables of all the ports $\langle n,\mathrm{out},h\rangle$ connected to port $\langle m,\mathrm{in},k\rangle$ by an edge.

Equations for output ports:

$$\sum_{e \in E[m,\mathrm{out},k]} \varphi_{\mathrm{e}} = 1 \quad \text{for all } \langle m,\ \mathrm{out},\ k\rangle \tag{12}$$

5.1.4. Objective function

$$\min_{x,y} \sum_{m \in M} (c_{\mathrm{fix},m} + c_{\mathrm{var},m}) \tag{13}$$

The above defined sets, variables, and Eqs. (9)–(13) together define the basic GDP representation (BGR).

Eq. (12) expresses the requirement that the sum of fractions is unity. Naturally, these fractions are non-negative, and limited by 1. This is expressed by Eq. (9).

Eqs. (10a) and (10b) expresses the feasibility constraints and cost functions attributed to the unit operations. For the sake of simplicity, negative variables are excluded. Any negative number can be expressed as a difference of two non-negative numbers. $\mathbf{P}_{\mathrm{t}}$ defines the unit operation together with its equipment (construction). Generally, subequation system $\mathbf{P}_{\mathrm{t}}$ includes components of both equality and inequality form. For example, a material balance around a unit is an equality, whereas a subequation expressing that a variable is greater than some minimum, as a complicated function of the other variables, is an inequality. Any equality can be expressed as a pair of two inequalities. For the sake of simplicity in notation and the proofs, "smaller than or equal to" relation is used here for $\mathbf{P}_{\mathrm{t}}$ in Eqs. (10a) and (10b). On the other hand, all our results remain valid if equality may occur in components of $\mathbf{P}_{\mathrm{t}}$; application of equality, if possible, is simpler in practice.

The objective function (Eq. (13)) is a sum of objective parts attributed to each unit. These parts are computed by P_{fix} and P_{var}. P_{fix} is the part of fix costs, depending on the design and control variables. P_{var} expresses the variable costs. Eq. (10a) is applied to the permanent units; Eq. (10b) is applied to the conditional units. Variable z in Eq. (10b) expresses the existence or non-existence of the unit. If the unit exists, the variables should satisfy the same equations that occur in Eq. (10a) ($\mathbf{P}_{\mathrm{t}}$, P_{fix}, P_{var}). When the unit does not exist, the cost increments must be zero, and all the other variables are also set to zero.

In Eq. (10b) logical relations $\wedge$ ('and') and $\vee$ ('or') and $\neg$ ('not') take place; this is a logical truth function. Each conditional unit is defined in the so-called disjunctive normal form; that is why this model is called GDP (disjunctive programming).

In our model the output ports behave as stream splitters (Eq. (12)); the input ports behave as stream unifiers (Eqs. (11a) and (11b)). In the unifiers the extensive variables are added together (Eq. (11a)), while the intensive variables of the unified stream are more complicated functions of all the stream properties (Eq. (11b)).

A significant difference between our BGR and the GDP model of Yeomans and Grossmann (1999) is that we do not provide additional logical relations between the logical variables (see Eqs. (4) and (18) in Yeomans & Grossmann, 1999) that would look in our case as:

$$\Omega(z) = \text{True} \tag{14}$$

There are two reasons for this change. Eq. (14) would be used for expressing possible substructures, and, based on engineering considerations, for excluding some not considered substructures from the set of structures. In our case, however, the supergraph is an R-graph, therefore no additional logical relations are needed to define the substructures. Instead, Eqs. (11a), (11b) and (12) express the splitter and unifier properties. On the other hand, we are automatically constructing here a basic representation, and do not include engineering considerations, that cannot be graphically represented. Such considerations will be included in a later phase of the modelling and design procedure.

Another significant difference is that we do not insert the P_{var} and P_{fix} functions directly into the objective function, but refer to them through the cost variables c_{var} and c_{fix}. Although the significance of this choice is not evident in the first sight, it will be transparent when the MINLP representation is formed. The essential consequence of this small change is that the binary variables in our model are not present in the objective function.

A third difference is that the assignment of permanent units is used by Yeomans and Grossmann for arbitrarily constraining the set of feasible structures. In BGR, only those units are assigned as being permanent that really are present in all the subgraphs. This is an unambiguous assignment.

5.2. Example 1—Basic GDP representation

Here, we construct the GDP representation of example problem 1 given in Table 1 and Figs. 7–12.

The continuous variables represent the inlet and outlet flow rates and the costs. There are not intensive nor other extensive variables of streams. Neither design or operation variables of units are applied.

Continuous variables:

$x_{m,\text{in}}$	inlet flow rate (tonnes/h)
$x_{m,\text{out}}$	outlet flow rate (tonnes/h)
$c_{m,\text{fix}}$	fix costs due to the existence of unit m
$c_{m,\text{var}}$	variable costs belonging to unit m

The permanent units are Units 5 and 6 because they belong to the only possible product. All the other units are conditional because either one of Units 1 or 4 may play the role of a single feed, and either Units 2 or 3 may be present in the solution without the other one. A z_m logical variable is also defined to each conditional unit. Finally, a continuous variable φ_e is assigned to each edge. There are altogether 27 continuous real variables (x_i, $i = 1, 2, \ldots 27$), including all the $x_{m,\text{in}}$, $x_{m,\text{out}}$, $c_{m,\text{fix}}$, $c_{m,\text{var}}$, and φ_e variables, and 4 logical variables (z_m, $m = 1, 2, 3, 4$). These variables can be lumped in an array **x** of the continuous variables, and an array **z** of the logical variables. Once these variables are assigned, their bounds can be defined. The bounds assign a range **X** of the continuous array, and a range **Z** of the logical variables, as in Eq. (15).

Now, the relations can be given as follow.

$$\mathbf{X} = \left\{ \begin{bmatrix} x_{1,\text{out}} \\ x_{2,\text{in}} \\ x_{2,\text{out}} \\ x_{3,\text{in}} \\ x_{3,\text{out}} \\ x_{4,\text{out}} \\ x_{5,\text{in}} \\ x_{5,\text{out}} \\ x_{6,\text{in}} \\ c_{\text{fix},1} \\ c_{\text{fix},2} \\ c_{\text{fix},3} \\ c_{\text{fix},4} \\ c_{\text{fix},5} \\ c_{\text{fix},6} \\ c_{\text{var},1} \\ c_{\text{var},2} \\ c_{\text{var},3} \\ c_{\text{var},4} \\ c_{\text{var},5} \\ c_{\text{var},6} \\ \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ \varphi_4 \\ \varphi_5 \\ \varphi_6 \end{bmatrix} \middle| \mathbf{x} \in R^{27}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -13 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \le \begin{bmatrix} x_{1,\text{out}} \\ x_{2,\text{in}} \\ x_{2,\text{out}} \\ x_{3,\text{in}} \\ x_{3,\text{out}} \\ x_{4,\text{out}} \\ x_{5,\text{in}} \\ x_{5,\text{out}} \\ x_{6,\text{in}} \\ c_{\text{fix},1} \\ c_{\text{fix},2} \\ c_{\text{fix},3} \\ c_{\text{fix},4} \\ c_{\text{fix},5} \\ c_{\text{fix},6} \\ c_{\text{var},1} \\ c_{\text{var},2} \\ c_{\text{var},3} \\ c_{\text{var},4} \\ c_{\text{var},5} \\ c_{\text{var},6} \\ \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ \varphi_4 \\ \varphi_5 \\ \varphi_6 \end{bmatrix} \le \begin{bmatrix} 3.57 \\ 2.04 \\ 5 \\ 1.53 \\ 1.11 \\ 1.11 \\ 1.11 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1.5 \\ 0 \\ 3.5 \\ 0 \\ 6.42 \\ 5 \\ 2.31 \\ 7.78 \\ 2 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right\}$$

$$\cup \mathbf{Z} = \left\{ \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} \middle| z_m \in \{\text{true}, \text{false}\} \right\} \tag{15}$$

Unit relations of permanent units:

Unit 5:

$$\left.\begin{array}{l} x_{5,\text{in}} > 0;\ x_{5,\text{out}} > 0 \\ x_{5,\text{out}} - 0.9\,x_{5,\text{in}} \le 0 \\ -x_{5,\text{out}} + 0.9\,x_{5,\text{in}} \le 0 \\ c_{\text{fix},5} = 3.5 \\ c_{\text{var},5} = 2\,x_{5,\text{out}} \end{array}\right] \tag{16a}$$

Unit 6:

$$\left.\begin{array}{l} x_{6,\text{in}} > 0 \\ c_{\text{fix},6} = 0 \\ c_{\text{var},6} = -13x_{6,\text{in}} \end{array}\right] \quad (16\text{b})$$

Unit relations of conditional units:
Unit 1:

$$\begin{bmatrix} z_1 \\ \wedge \\ x_{1,\text{out}} > 0 \\ c_{\text{fix},1} = 0 \\ c_{\text{var},1} = 1.8x_{1,\text{out}} \end{bmatrix} \vee \begin{bmatrix} \neg z_1 \\ \wedge \\ x_{1,\text{out}} = 0 \\ c_{\text{fix},1} = 0 \\ c_{\text{var},1} = 0 \end{bmatrix} \quad (16\text{c})$$

Unit 2:

$$\begin{bmatrix} z_2 \\ \wedge \\ x_{2,\text{in}} > 0;\ x_{2,\text{out}} > 0 \\ x_{2,\text{out}} - \ln(x_{2,E} + 1) \le 0 \\ -x_{2,\text{out}} + \ln(x_{2,\text{in}} + 1) \le 0 \\ c_{\text{fix},2} = 1 \\ c_{\text{var},2} = x_{2,\text{out}} \end{bmatrix} \vee \begin{bmatrix} \neg z_2 \\ \wedge \\ x_{2,\text{in}} = 0;\ x_{2,\text{out}} = 0 \\ c_{\text{fix},2} = 0 \\ c_{\text{var},2} = 0 \end{bmatrix} \quad (16\text{d})$$

Unit 3:

$$\begin{bmatrix} z_3 \\ \wedge \\ x_{3,\text{in}} > 0;\ x_{3,\text{out}} > 0 \\ x_{3,\text{out}} - 1.2\ \ln(x_{3,\text{in}} + 1) \le 0 \\ -x_{3,\text{out}} + 1.2\ \ln(x_{3,\text{in}} + 1) \le 0 \\ c_{\text{fix},3} = 1.5 \\ c_{\text{var},3} = 1.2\,x_{3,\text{out}} \end{bmatrix} \vee \begin{bmatrix} \neg z_3 \\ \wedge \\ x_{3,\text{in}} = 0;\ x_{3,\text{out}} = 0 \\ c_{\text{fix},3} = 0 \\ c_{\text{var},3} = 0 \end{bmatrix} \quad (16\text{e})$$

Unit 4:

$$\begin{bmatrix} z_4 \\ \wedge \\ x_{4,\text{out}} > 0 \\ c_{\text{fix},4} = 0 \\ c_{var,4} = 7\,x_{4,\text{out}} \end{bmatrix} \vee \begin{bmatrix} \neg z_4 \\ \wedge \\ x_{4,\text{out}} = 0 \\ c_{\text{fix},4} = 0 \\ c_{\text{var},4} = 0 \end{bmatrix} \quad (16\text{f})$$

Source and sink units do not have operation equations. The operation equations of the other units are formed as inequalities. (These relations can be written in equation form in the MINLP representation.)

Equations for input ports:

$$\left.\begin{array}{l} x_{2,\text{in}} = \varphi_1 \cdot x_{1,\text{out}} \\ x_{3,\text{in}} = \varphi_2 \cdot x_{1,\text{out}} \\ x_{5,\text{in}} = \varphi_3 \cdot x_{2,\text{out}} + \varphi_4 \cdot x_{3,\text{out}} + \varphi_5 \cdot x_{4,\text{out}} \\ x_{6,\text{in}} = \varphi_6 \cdot x_{5,\text{out}} \end{array}\right. \quad (17)$$

Equations for output ports:

$$\left.\begin{array}{l} \varphi_1 + \varphi_2 = 1 \\ \varphi_3 = 1 \\ \varphi_4 = 1 \\ \varphi_5 = 1 \\ \varphi_6 = 1 \end{array}\right. \quad (18)$$

Those φ_e variables, which have value 1, may be dropped in an MINLP representation different from BGR and BMR.

5.2.1. Objective function

$$\min_{x,y} \sum_{i=1}^{6} (c_{\text{fix},i} + c_{var,i}) \quad (19)$$

5.3. Basic MINLP representation

BGR can be solved using appropriate algorithms utilizing the GDP form; but representing the synthesis problem by MINLP model is yet more common and well-spread in the chemical engineering communities. However, well formulating a synthesis problem as MINLP is difficult, whereas formulating it as a GDP is most convenient. Respectably, it is worth *automatically transforming* the conveniently formulated BGR of the synthesis problem to its MINLP formulation.

BMR is automatically formed from BGR by transforming the disjunctions of Eq. (10b) into algebraic form while substituting binary variables y in the place of the logical variable z. The MINLP problems are usually solved using decomposition methods, and for this aim the binary variables are applied in linear constraints, cf. Eqs. (5) and (7). Several ways can be applied for performing such a transformation. Widespread transformations are the so-called 'Big M', 'Multi M' and 'Convex Hull' methods (Grossmann and Turkay, 1996; Raman and Grossmann, 1991; Szitkai, Lelkes, Rev, & Fonyo, 2003; Vecchietti, Lee, & Grossmann, 2003). Any of these methods may be used for defining BMR; all are equally proper methodologies. For example, the Big M transformation of Eq. (10b) leads to the following subsystem of equations:

$$\mathbf{x}_{m,\text{in}} \le \mathbf{U}_{m,\text{in}} y_m \quad (20)$$

$$\boldsymbol{\varepsilon} - \mathbf{x}_{m,\text{in}} \le \mathbf{U}_{m,\text{in}}(1 - y_m) \quad (21)$$

$$\mathbf{x}_{m,\text{out}} \le \mathbf{U}_{m,\text{out}} y_m \quad (22)$$

$$\boldsymbol{\varepsilon} - \mathbf{x}_{m,\text{out}} \le \mathbf{U}_{m,\text{out}}(1 - y_m) \quad (23)$$

$$0 \le \mathbf{o}_m \le \mathbf{U}_{m,o} y_m \quad (24)$$

$$0 \le \mathbf{d}_m \le \mathbf{U}_{m,d} y_m \quad (25)$$

$$0 \le c_{\text{fix},m} \le U_{m,\text{fix}} y_m \quad (26)$$

$$L_{m,var} y_m \le c_{var,m} \le U_{m,var} y_m \quad (27)$$

$$\mathbf{L}_{m,P} y_m \le \mathbf{P}_t(\mathbf{x}_{m,\text{in}}, \mathbf{x}_{m,\text{out}}, \mathbf{d}_m, \mathbf{o}_m) \le \mathbf{U}_{m,P}(1-y_m) \quad (28)$$

$$L_{m,P_{\text{fix}}}(1-y_m) \le c_{\text{fix},m} - P_{\text{fix}_t}(\mathbf{d}_m) \le U_{m,P_{\text{fix}}}(1-y_m) \quad (29)$$

$$L_{m,P_{\text{var}}}(1-y_m) \le c_{var,m} - P_{\text{var}_t}(\mathbf{x}_{m,\text{in}}, \mathbf{x}_{m,\text{out}}, \mathbf{d}_m, \mathbf{o}_m) \le U_{m,P_{\text{var}}}(1-y_m) \quad (30)$$

where L and U are the lower and upper bounds, respectively, and $\boldsymbol{\varepsilon}$ is a non-zero vector of small length, introduced in order to exclude zero vector from the domain.

5.4. Example 1—Basic MINLP representation

Once the GDP unit relations of the conditional units are given (Eq. (16c)–(16f)), the *basic MINLP representation* can be automatically generated. First, the logical variables z are transformed into binary y ones; thus, the range $\mathbf{Y}$ of the binary variables becomes:

$$\mathbf{Y} = \left\{ \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \middle| \mathbf{y} \in \{0,1\}^4 \right\} \quad (31)$$

Then, the equations can also be transformed, accordingly. The transformed equations are listed below:

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

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

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

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

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

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

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

Some of these equations can be written in simpler form. For example, a simpler form of the equations concerned to $c_{\text{fix},4}$ (from Eqs. (34) and (37)) is the following:

$$c_{\text{fix},4} = 0 \quad (39)$$

Instead of the four inequalities in Eq. (36), two inequalities are enough:

$$\begin{aligned} &-1.11 y_2 \le x_{2,\text{out}} - \ln(x_{2,\text{in}}+1) \le 1.11(1-y_2) \\ &-1.11 y_3 \le x_{3,\text{out}} - 1.2\ \ln(x_{3,\text{in}}+1) \le 1.11(1-y_3) \end{aligned} \quad (40)$$

These simpler forms (Eqs. (39) and (40)) are, however, not used here in order to ensure the unambiguous form of the BMR. The non-transformed variables and the non-transformed equations are unchanged; therefore they are not shown here.

6. MINLP representation

For avoiding multiplicity and redundancy in the MINLP problem representation, one have to be able to determine if a given MR represents a supergraph (or a superstructure) or not. The same question emerges if two representations have to be compared according to their feasible regions. Without special care, one cannot be sure if the MR really represents all the considered structures. We have not find in the literature such a definition of MR that could be applied to solve this problem.

The main difficulty here lies in the fact that the number of variables, and even their names and characteristics are subject to arbitrary variations.

Here, we suggest defining the MR through a fixed form of BMR. There is a double merit of this definition. First, in this way the question of representation can be solved, as shown below. Second, such a BMR can automatically be generated and can serve as a reference representation.

MR should be defined in such a way that each of its solutions unambiguously assigns a graph state and thus a flowsheet in the same way as BMR does, but it may contain

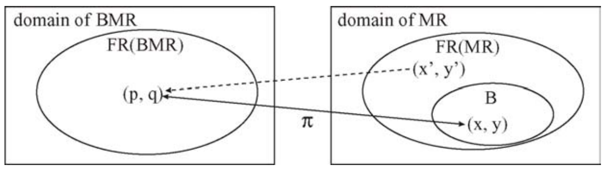

Fig. 13. The mapping π, and the relation between FR(BMR) and FR(MR).

arbitrary superfluous information. This may include superfluous information on structures not considered, or it may also include redundant information on the considered structures.

Here, we define MR in two different, but related, ways. The first, general, definition deals with the ability of an MINLP problem formulation to represent the considered structures of the synthesis task. This general definition includes the existence of a bijection π between FR(BMR) and a subset B of FR(MR). This definition can be applied to check if an MINLP problem formulation can anyhow represent the considered structures of a synthesis task.

The general definition of MR is the following:

An MR (MINLP problem representation) represents a graph if the following two conditions are satisfied:

1. Such a subregion $B \subseteq FR(MR)$ exists that a bijection π: $B \Leftrightarrow FR(BMR)$ can be given where FR(BMR) is the feasible region of the basic representation of the graph, and FR(MR) is the feasible region of the actual MR in question.
2. For each solution (x,y) of MR that lies in B (that is $(x,y) \in B$) $Z_{MR}(x,y)$ is equal to $Z_{BMR}(\pi(x,y))$, where Z_{MR} and Z_{BMR} denote the objective functions of MR and BMR, respectively.

This general definition of MR is explained in Fig. 13. The constraining conditions, that enforce the solution to represent an R-graph, assign the feasible region FR(BMR). Points of FR(BMR) describe the subgraphs (but not necessarily just the considered graphs). B is a subset of the feasible region of MR, that is, it is a subset of FR(MR). If a part of B was not subset of FR(MR) then some of the feasible solutions of BMR, and thus some of the subgraphs, would not be represented by MR. As B is a subset of FR(MR), the feasible solutions of MR include all the solutions that are mapped to the states of the subgraphs. The connections between the representations and sets are shown in Fig. 14.

Here, we remind the reader to what was written in the paragraph below Eq. (14). The superstructure is represented by a supergraph, and no structural constraints additional to what is applied in BGR are applied in BMR. Therefore, FR(BMR)

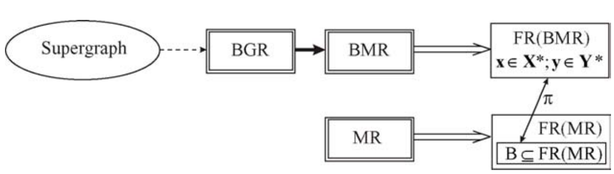

Fig. 14. Connections between sets and representations.

includes all the subgraphs of the supergraph, not just the considered graphs. Thus, all the subgraphs of the supergraph are represented by FR(BMR) and by $B \subseteq FR(MR)$. It then follows that if an R-graph is represented by an MR then all the subgraphs of that R-graph are also represented by that MR.

According to the definition of isomorphy, bijection can always be given between two isomorphic graphs. It then follows that if a graph is represented by an MR then all its isomorphic graphs are also represented by that MR.

FR(MR) may contain feasible solutions outside of B, and these solutions may correspond to graphs and structures not included in FR(BMR). Such a correspondance can be expressed by some mapping from FR(MR) to the set of some flowsheets, or their representing graphs, including all those represented by FR(BMR), and others not represented by it. That is, an arbitrary representation of the synthesis task can be wider than the BMR, but cannot be narrower. In this way a representation can carry any additional information, but it also carries all the information necessary for describing the subgraphs of the supergraph.

The second, restricted, definition of MR includes a particularly assigned surjection ψ from FR(MR) to FR(BMR). Such a mapping ψ can be assigned only if the above bijection π exists. Once such a surjection ψ is given, a bijection π with the property $\pi \subseteq \psi$ always exists. Thus, this representation can be defined as a restriction of the general MR to the application of a particularly selected surjection ψ.

MR may contain a feasible solution (x',y'), outside of B, that is also mapped to (p,q) according to the surjection ψ, as is also shown in Fig. 13. In this case the MR with the particular selection of ψ is redundant, because two feasible solutions are mapped to the same flowsheet. Such a redundancy is not excluded, and avoiding this kind of redundancy is not always preferable.

This restricted definition of MR is beneficial in defining ideal MINLP representation, and is utilized in Part II. Which definition is applied must be clear from the textual environment.

We cannot emphasize with great enough weight that the actual MINLP problem formulation of MR is not in any way bound to the variables and equations of BMR. The only restriction is the existence of a bijection between a subset $B \subset FR(MR)$ and FR(BMR) in such a way as to provide with the same objective value. The engineer has the freedom to apply a formulation best fitting to convenience and numerical efficiency.

6.1. Example 1—An MINLP representation

Kocis and Grossmann (1987) presented an MINLP representation to a planning problem identical to our example. This MR is also presented in Appendix B. Here, we demonstrate that the MR of Kocis and Grossmann (1987) represents the same superstructure as our BMR does for Example 1 presented in Section 3 and also dealt with in Section 5.2. For this aim, we have to give a bijective mapping from subregion B of FR(MR) to FR(BMR).

We show how this mapping can be constructed. First consider the source of raw material B that is represented by continuous variable b_1 in the network of Kocis and Grossmann (Fig. 8), and by a pair of continuous variable $x_{4,\mathrm{out}}$ and the binary variable y_4 in our basic MINLP representation. Variable $x_{4,\mathrm{out}}$ describes the flow rate of the material flow if that raw material is applied. Whether raw material B is applied is described by the binary variable y_4; this formally corresponds to the existence of Unit 4 (source unit). Here, we give a mapping between the values of the pair (y_4, $x_{4,\mathrm{out}}$) falling in the feasible region of BMR and the values of b_1 falling in the B subset of the feasible region of MR. (If MR represents all the structures represented by BMR then all the feasible values of (y_4, $x_{4,\mathrm{out}}$) should have a picture value b_1, and each such picture value should have an *unambiguous* (y_4, $x_{4,\mathrm{out}}$) anchestor value. On the other hand, the b_1 values in FR(MR) outside of B do not have ancestor according to this mapping.)

In the MR, b_1 is a non-negative variable without any upper bound; thus, b_1 can take any non-negative value. But the set of feasible values of b_1 is narrower. This feasible set can be determined as follows. There is an upper bound for the product c:

$$c \le c^{\mathrm{UP}} = 1 \tag{41}$$

The operation equation of Unit III is:

$$c - 0.9b = 0 \tag{42}$$

From here, the maximal value that b can take is

$$b^{\max} = \frac{c^{\mathrm{UP}}}{0.9} = 1.11 \tag{43}$$

The material balance of mixing b_i flows is

$$b_1 + b_2 + b_3 - b = 0 \tag{44}$$

The values of either b_2 or b_3 can be arbitrary near zero. Therefore, the maximal value that b_1 can take is almost the maximal value of b. For practical purposes, we take the limit:

$$b_1^{\max} = b^{\max} = 1.11 \tag{45}$$

The feasible set of b_1 is, therefore, the closed interval [0,1.11] (see Fig. 15).

The lower and upper bounds of the variables in the graph representation were determined in Eqs. (15) and (31). For y_4 and $x_{4,\mathrm{out}}$ variables these are:

$$\begin{aligned} &y_4 \in \{0, 1\} \\ &x_{4,\mathrm{out}} \in [0, 1.11] \end{aligned} \tag{46}$$

However, there are additional constraints for these variables in BMR in order to exclude some extreme and prohibited cases:

$$x_{4,\mathrm{out}} \le 1.11 y_4 \tag{47a}$$

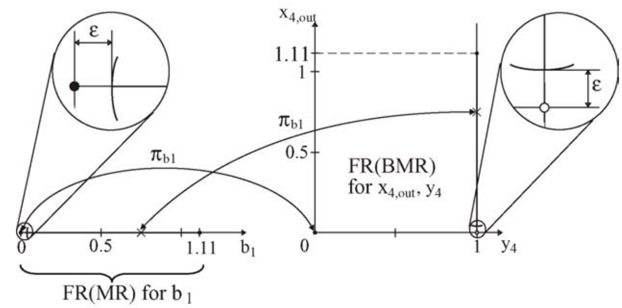

Fig. 15. Bijection π applied to the feasible values of variable b_1.

$$-x_{4,\mathrm{out}} \le 1.11(1 - y_4) - \varepsilon \tag{47b}$$

Eqs. (46), (47a) and (47b) describe the corresponding subset, belonging to these variables in BMR, of the feasible set of FR(BMR).

A bijective mapping π_{b_1} between the feasible sets of b_1 and (y_4, $x_{4,\mathrm{out}}$) can be defined as given by the analytical definition Eq. (48) and shown in Fig. 15:

$$\pi_{b1} : \begin{cases} \text{if } b_1 = 0 \Rightarrow x_{4,\mathrm{out}} = b_1 \text{ and } y_4 = 0 \\ \text{if } \varepsilon \le b_1 \le 1.11 \Rightarrow x_{4,\mathrm{out}} = b_1 \text{ and } y_4 = 1 \end{cases} \tag{48}$$

In bijection π_{b_1}, variable b_1 can take values from region $\mathrm{B}_{b_1} = \{0;[\varepsilon,1.11]\}$, that is the union of the point of the exact zero and of the closed section between ε and 1.11. This set B_{b_1} is the subregion of the feasible set of b_1 in FR(MR). For all the other variables in MR of Kocis and Grossmann, a similar bijection can be constructed. In this way it can be proved that the MINLP representation of Kocis and Grossmann represents the same superstructure as represented by the basic GDP representation.

7. Multiplicity of the MINLP representation

Depending on the actual form of the MINLP problem, multiple local optima, and even multiple global optima may occur. 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. Most of the solution methodologies (e.g. Branch & Bound, Generalised Benders Decomposition, etc.) apply forming continuous (NLP) subproblems combined with either MILP subproblems or tree enumeration. Therefore, making distinction between binary and continuous multiplicities seems useful.

One way of forming NLP subproblems is fixing all the binary values. (For example, this is applied in the Outer Approximation methods, see, e.g. Duran & Grossmann, 1986a,b). We will speak of *binary multiplicity* if two different NLP subproblems, belonging to differently fixed values of the binary variables, lead to the same optimum. The continuous variables may also take different values.

On the other hand, any NLP subproblem can also have several different optimal solutions with equal objective values.

This means that different values of the continuous variables of the NLP, assigned by fixed values of the binary variables, lead to the same optimum. We say that the MINLP problem has *continuous multiplicity* if one of its NLP problems has this property.

Both the binary and the continuous multiplicity are determined by the form of the MINLP problem. The actual form is somewhat arbitrary; it is in the hand of the engineer or the mathematician. A great extent of multiplicity can be introduced or removed by inapt or efficient formulation. A generally applied way of formulation is assigning binary variables for describing existence or non-existence of units. This conventional methodology is applied in the definition of BGR. If this kind of binary variables is applied and if there is redundancy in the superstructure (irrespectively if it is represented by graph or not) then a great extent of binary multiplicity, originated from structural multiplicity and redundancy, may occur. In some cases this kind of binary multiplicity is evident, in other cases it is not, and in some cases they are even not recognized. This is also manifested in the works of the researchers who tried to eliminate the multiplicity, e.g. Lelkes, Szitkai, Rev, and Fonyo (1991), Reneaume, Heritier, Domenech, and Joulia (1995), Reyes-Labarta and Grossmann (2001). Of course, some sources of binary multiplicity may also be independent of the superstructure.

If a supergraph is structurally redundant (i.e. if there are isomorphic graphs amongst its subgraphs) then BMR has binary multiplicity.

This statement can easily be proved: Let a supergraph R be structurally redundant. Then it has at least two isomorphic subgraphs, R_1 and R_2. Since R_1 and R_2 are isomorphic, they represent the same structure. BMR represents all the subgraphs of R, including R_1 and R_2. R_1 can be assigned by fixing the binary variables to value y_1, and R_2 can be assigned by fixing the binary variables to $y_2 \neq y_1$. These different binary values assign different NLP-s, but their optima are equal since R_1 and R_2 represent the same structure. Thus BMR has binary multiplicity.

Conversely, the structural redundancy of the supergraph does not follow from the binary multiplicity of its BMR. It is easy to construct two non-isomorphic graphs and an objective function that leads to the same optimum.

A small, perhaps irrealistic, counter-example is shown in Fig. 16. Here the process has a single input and a single output stream, their measure are denoted by the real variables x and y, respectively. The single source unit and the single sink unit are permanent units of the supergraph. The two other units are applied parallel; therefore, either one of them may be omitted, leading to different subgraphs, as is shown in Fig. 17. Suppose that simultaneous existence of Units 2 and 3 are prohibited.

The two parallel units are of different type; therefore, these two subgraphs are not isomorphic; they represent different process structures. There is no structural redundancy here. Let the existence or omitting of the parallel units be described by binary variables.

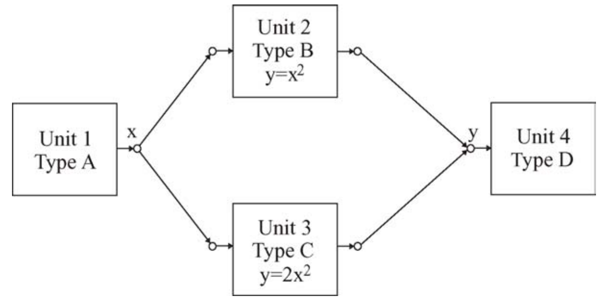

Fig. 16. Supergraph of the small counter-example.

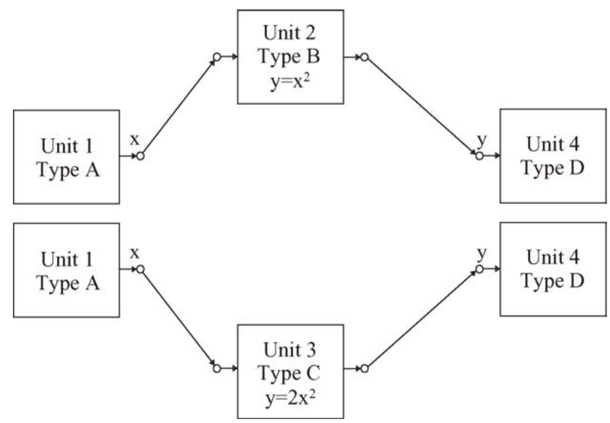

Fig. 17. Two non-isomorphic feasible subgraphs of the supergraph.

Let the objective function be simply y, and let this objective be determined by the simple formulas shown in Figs. 16 and 17, according to which parallel unit is in use. In case of Unit 2: $y = x^2$; in case of Unit 3: $y = 2x^2$. In case of $x = 0$, y has the same value ($y = 0$) in both cases. This is a simple case of binary multiplicity without having structural redundancy.

As is written by Rev et al. (submitted for publication), bypass multiplicity may occur if not R-graphs are applied. Bypass multiplicity is manifested via continuous multiplicity in the MINLP representation. On the other hand, application of R-graphs does not eliminate all the continuous multiplicity originated from the superstructure. It is an open question if a superstructure eliminating all such multiplicity can always be given.

In Part II versions of MR, and ways they can be constructed, are shown with the property of decreased multiplicity and redundancy. In that part, the effect of these phenomena on the solution chances will also be analysed.

8. Conclusion

A definition of MINLP representation (MR), of process synthesis tasks, conform with the R-graph representation is developed. The essential problems in mathematically defining the MR are representativeness and uniqueness. Whether an MR represents a superstructure and all its considered substructures should be determinable. An other crucial problem is comparison of two different MINLP representations to decide if some process flowsheets are represented by both of them.

The MINLP representation (MR) is to be based on graph representation of structures. Since the same structure can be represented by several isomorphic graphs, and since this isomorphism is a serious source of redundancy and multiplicity, isomorphic graphs should not be present in the set of considered graphs, and are to be excluded from the represented feasible domain of MR.

In order to avoid ambiguousity in defining the variables of the MR, we first defined a basic GDP representation (BGR) that involves logical relations. BGR can be constructed by applying a standard 'naïve' representation of the process. On this base, a so-called basic MINLP representation (BMR) can automatically be constructed by unambiguously transforming the logical relations to algebraic ones. In this way BMR can serve as a reference representation. BGR and BMR are the so-called basic representations of the synthesis problem. The actual solution of BMR unambiguously assigns a subgraph of the supergraphs, and finally the flowsheet itself.

The BMR constructed in this way may contain, and usually does contain, multiplicity and redundancy. Constraints on the feasible domain, for excluding isomorphism, and for decreasing multiplicity and redundancy, are to be introduced in a later phase of model construction. However, introduction of any additional constraints, and introduction of new variables, makes it questionable if the modified MR represents all the considered graphs, and especially the optimal one.

MR is defined through a fixed form of BMR. In this way equivalency and representativeness of MR-s in general form can be analyzed by reducing them to their BMR and by comparing their feasible domains. BMR can be automatically generated and can serve as a reference representation. Construction of the supergraph, assignment of considered graphs, and construction of BGR, BMR, and MR are demonstrated.

Binary and continuous multiplicity of MR are defined. If an R-graph is structurally redundant (i.e. there are isomorphic graphs amongst their subgraphs) then BMR has binary multiplicity, as is proven here. Conversely, the structural redundancy of the supergraph does not follow from the binary multiplicity of its BMR.

Different kinds of multiplicity and redundancy measures of the MINLP representation will be defined in Part II of this series to help inventing tools for decreasing their detrimental effect. Alternative MINLP representations will there be examined and compared from the viewpoint of ideality, minimality, and solution properties.

Acknowledgement

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

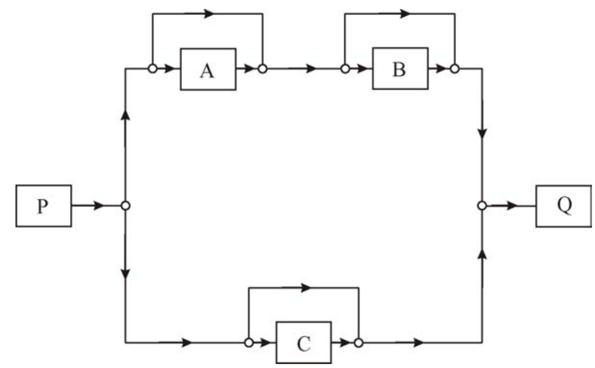

Fig. A1. Branched structure with by-passes.

Appendix A. By-pass redundancy

Fig. A1 serves as a simple (and arbitrary) example for demonstrating the redundancy related to by-passes. (Similar figure can be seen, e.g. in Reneaume et al. (1995).) By-passing Unit A is necessary to let Unit B exist even if A is not included in the structure. By-passing Unit B is necessary for the reverse reasoning. An identical substructure at the lower branch may occur. For our didactical purposes we apply just a Unit C there, also bypassed. These units and streams form the superstructure. Exclusion of a unit from the final structure does not exclude its by-pass stream. Thus, excluding Unit C may bring to life the structure shown in Fig. A2.

Let the flow rates of the streams leaving Units P, A, and B are given; let the flow rates of streams entering into Units A, B, and Q are also given. For the sake of simplicity and clarity, suppose that the flow rates x and y are equal: $x = y$. Then the sum $x + z$, equaling the sum $y + z$, is a constant, where z is the flow rate of the stream by-passing the upper branch. For example, let the flow rate leaving Unit P be 100, $z = 50$, $x = 20$, and the input to Unit A be 30. In the same time $y = 20$, and let the output from Unit B be 35, then the input to Unit Q is 105.

Then the flow rate z can be changed on the cost of simultaneously changing the values of x and y, without having any influence on the units. Without changing the input and output of Units P, A, B, and Q, the flow rates may be, for example, $z = 60$, $x = y = 10$, or $z = 30$, $x = y = 40$.

As a result, the objective function (not given here) may remain constant in a continuous subdomain of the feasible

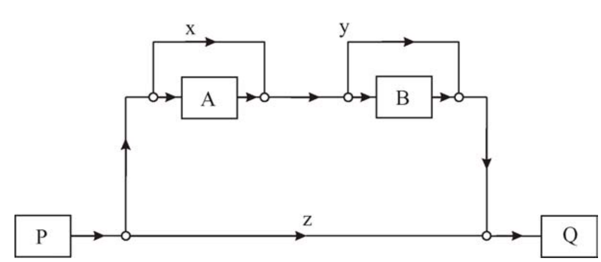

Fig. A2. A structure with redundant by-passes.

solutions. This phenomenon, that has a detrimental effect on optimizers, is called by-pass redundancy.

Appendix B. MINLP representation of Kocis and Grossmann (1987)

$$\min z = y_{\mathrm{I}} + 1.5y_{\mathrm{II}} + 3.5y_{\mathrm{III}} + 1.8a_1 + 7.0b_1 + b_2 + 1.2b_3 - 11.0c$$
$$\text{s.t.}\, b_2 - \ln(1 + a_2) = 0$$
$$b_3 - 1.2\ \ln(1 + a_3) = 0$$
$$c - 0.9b = 0$$
$$b_1 + b_2 + b_3 - b = 0$$
$$a_1 - a_2 - a_3 = 0$$
$$a_2 - 5y_{\mathrm{I}} \le 0$$
$$a_3 - 5y_{\mathrm{II}} \le 0$$
$$b - 5y_{\mathrm{III}} \le 0$$
$$c \le 1$$
$$b_2 \le 5$$
$$y_{\mathrm{I}}, y_{\mathrm{II}}, y_{\mathrm{III}} \in \{0, 1\}^3$$
$$a_1, a_2, a_3, b, b_1, b_2, b_3, c \ge 0$$

References

Duran, M. A., & Grossmann, I. E. (1986a). An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, *36*, 307–339.

Duran, M. A., & Grossmann, I. E. (1986b). A mixed-integer nonlinear programming algorithm for process systems synthesis. *Chemical Engineering Journal*, 592–596.

Friedler, F., Fan, L. T., & Imreh, B. (1998). Process network synthesis: Problem definition. *Networks*, *31*(2), 119–124.

Friedler, F., Tarjan, K., Huang, Y. W., & Fan, L. T. (1992). Graph-theoretic approach to process synthesis: Axioms and theorems. *Chemical Engineering Science*, *47*(8), 1973–1988.

Friedler, F., Tarjan, K., Huang, Y. W., & Fan, L. T. (1993). Graph-theoretic approach to process synthesis: Polynomial algorithm for maximal structure generation. *Computers and Chemical Engineering*, *17*(9), 929–942.

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

Grossmann, I. E., & Turkay, M. (1996). Solution of algebraic systems of disjunctive equations. *Computers and Chemical Engineering*, *20*, 339–S344.

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

Lee, S., & Grossmann, I. E. (2000). New algorithms for nonlinear generalized disjunctive programming. *Computers and Chemical Engineering*, *24*, 2125–2141.

Lelkes, Z., Szitkai, Z., Rev, E., & Fonyo, Z. (2000). Rigorous MINLP model for ethanol dehydration system. *Computers and Chemical Engineering*, *24*, 1331–1336.

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

Reneaume, J.M., Heritier, L., Domenech, S., & Joulia, X. (1995). Synthese optimale de reseaux d'echangeurs de chaleur dans l'environement d'un simulateur modulaire. 5éme Congrès Français de Génie des Procédés, Lyon, 19–21 Septembre 1995. Collection Récents Progrés en Génie des Procédés, Le Génie des Procédés Complexes. 9, 42, 435–440.

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

Reyes-Labarta, J.A., & Grossmann, I.E. (2001). Optimal synthesis of liquid–liquid multistage extractors. Proceedings of ESCAPE-11. 27–30 May, 2001. Kolding, Denmark; 487–492. Amsterdam: Elsevier.

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 and Chemical Engineering*, *26*(11), 1501–1516.

Vecchietti, A., Lee, S., & Grossmann, I. E. (2003). Modeling of discrete/continuous optimization problems: Characterization and formulation of disjunctions and their relaxations. *Computers and Chemical Engineering*, *27*(3), 433–448.

Yeomans, H., & Grossmann, I. E. (1999). A systematic modeling framework of superstructure optimization in process synthesis. *Computers and Chemical Engineering*, *23*(6), 709–731.