Structural Multiplicity and Redundancy in Chemical Process Synthesis with MINLP

Tivadar Farkas, Endre Rev, and Zoltan Lelkes
Chemical Engineering Department, Budapest University of Technology and Economics
H-1521, Budapest, Hungary

Abstract
How easy the solution of an MINLP model of a superstructure is usually analyzed according to the shape (linearity, convexity, relaxation, etc) of the equations, see e.g. Grossmann (1996). Here relations between the (super)structures and their MINLP representation are studied. In order to analyze this relation, we have defined ideal MINLP and binarily minimal MINLP representations. The effect of ideality and the number of binary variables on the solution time are compared on test examples. The first example is the synthesis problem of Kocis and Grossmann (1987). An ideal and, in the same time, binarily minimal MINLP representation has been constructed and solved. The second example is the membrane train of an industrial ethanol dehydration problem (Lelkes et al., 2000). The representations, solved on a Sun Sparc station using GAMS DICOPT++ solver, are compared according to the maximal size of solvable problems.

Keywords: multiplicity, ideality, redundancy, MINLP, representation

1. Introduction
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 they should include the minimum number of unconsidered structures, for technical reasons. This is not a problem of mathematics. Once all the considered structures are either explicitly known or implicitly given, a superstructure can be constructed using mathematics (Friedler et al., 1992, 1993, 1998). 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 optimization model of the superstructure may be an 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). How easy the solution of an MINLP model of a superstructure is has usually been analyzed according to the shape (linearity, convexity, relaxation, etc) of the equations, see e.g. Grossmann (1996). We think, however, that relations between the (super)structures and their MINLP representation is also to be systematically studied.

Superstructures, usually represented by graphs, may be redundant since their subgraphs may have multiplicity; and, consequently, their model may have different solutions representing the same process. Although some researchers apply intuitive methodology for decreasing the multiplicity, it is worth analyzing its sources and looking for ways to decrease their effect. Rev and Lelkes (2004) mathematically defined process flowsheet

* Author to whom correspondence should be addressed: lelkes@mail.bme.hu
superstructure, explained, visualized, and demonstrated with examples. The process flowsheet structure is defined as a class of isomorphic R-graphs, also defined there. Basic MINLP representation for reference purposes, as well as ideal and binarily minimal MINLP representations are defined (see the details in Lelkes et al., 2004), in order to analyze this relation. The effect of ideality and of the number of binary variables on the solution time and size of solvable problems are compared on examples.

2. Considered structures and graphs

To achieve the targeted chemical process, the structure can usually be selected from an infinite number of physically feasible process structures, 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, either explicitly or implicitly, a finite set of process structures to be studied, here called considered structures. Once this set 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 set of variables and mathematical relations. The graph representation of the superstructure will here be called the supergraph. Structural multiplicity is an important phenomenon caused by the possibility of representing the same structure with different graphs. 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). See, as an over-simplified example, Figs. 1ab. The R-graph of the superstructure is shown in Fig. 1a. In the R-graph representation, the nodes represent units with their input and output ports; the edges represent the streams connecting the ports. This graph has several subgraphs, representing substructures of the superstructure, two of which are shown in Fig. 1b. The types of Unit 2 and Unit 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 labeled according to mathematical definition of graphs. Consequently, the structures are not equivalent to graphs but to sets of isomorphic graphs (see Rev and Lelkes, 2004). A structure can be represented by any of these isomorphic graphs. The isomorphic graphs are not necessarily represented distinctly with a mathematical model, because they correspond to the same process structure.

Since the detrimental effect of structural multiplicity and redundancy can be decreased by not 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.
3. MINLP representations

In order to generate MINLP representation (MR) reliably representing all the considered graphs, we first defined (Lelkes et al., 2004) basic representations that can be constructed in a standard manner, and then defined MR-s with feasible domain mappable to those of the basic representations. A General Disjunctive Programming (GDP) model is first constructed in the spirit of Yeomans and Grossmann (1999) but based on the R-graph representation. The disjunctions of the GDP model can be transformed into algebraic form while substituting binary variables in the place of logical variables. This automatically formed representation called Basic MINLP Representation (BMR). It does not contain any additional constraints than the GDP representation. BMR represents the graphs of all the physically feasible structures. Therefore isomorphic graphs, i.e. not considered graphs are also represented.

An MR is called Ideal MINLP Representation (IMR) if it represents all the considered graphs, and no other graph. Non-ideality of MR can be measured by \( N = (m-n)/n \); where \( m \) and \( n \) are the number of represented and considered graphs, respectively. Problems with integer or binary variables (ILP/MILP/MINLP) are usually more difficult to solve than those without them (LP/NLP). Generally the difficulty (solution time, for example) of solving the problem drastically increases with the number of integer variables, supposing equal relaxation effects. Therefore, decreasing the number of such variables has a key role in increasing the scale of the solvable problems and reducing the solution work. A representation is called Binarily Minimal MINLP Representation (BMMR) if it applies a minimum number of binary variables used to make distinction between the represented graphs, supposing this distinction is made with the use of binary variables. (In ideal case, all, and only, the represented graphs are considered.) A vector 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 represented graphs: \( 2^n \geq k \). That is, \( n \) is the smallest whole number that satisfies \( n \geq \log_2 k \). The effect of ideality and the number of binary variables on the solution time are compared on test examples.

4. Example 1 – Synthesis problem

A small planning problem (Kocis and 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. 2 using an OTOE network for visualising the superstructure. This figure does not represent a graph in mathematical sense. For the sake of exact discussion, the R-graph representation of the problem will be used instead, see Fig. 3. Unit 1 and Unit 4 are the sources of raw materials A and B; Unit 6 is the sink of product C.

Fig. 2. Example network  
Fig. 3. R-graph representation of Example 1
The MR originally given by Kocis and Grossmann (MR of Kocis) is solved for reference. Another MINLP representation (MR) was also constructed via the BMR of the problem, based on R-graph representation, and then excluding the redundant and unnecessary equations. Then, 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. These four representations were solved on a Sun Sparc workstation, using GAMS DICOPT++ solver (Brooke et al., 1988). All the binary variables were assigned the initial value 0.5 in each case. The solution times and results are collected in Table 1. The number of iterations shows how many main (outer) iterations were done. The solution time is given in CPU sec. The same optimum, shown in Fig. 4, was found in each case.

Table 1. Solution times of Example 1

<table>
<thead>
<tr>
<th>representation</th>
<th>objective function</th>
<th>number of iterations</th>
<th>solution time (s)</th>
<th>NLP (s)</th>
<th>MILP (s)</th>
<th>NLP/it. (s)</th>
<th>MILP/it. (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MR of Kocis</td>
<td>-1.923</td>
<td>9</td>
<td>0.98</td>
<td>0.61</td>
<td>0.37</td>
<td>0.068</td>
<td>0.041</td>
</tr>
<tr>
<td>MR</td>
<td>-1.923</td>
<td>17</td>
<td>2.37</td>
<td>1.18</td>
<td>1.19</td>
<td>0.069</td>
<td>0.070</td>
</tr>
<tr>
<td>IMR</td>
<td>-1.923</td>
<td>13</td>
<td>1.70</td>
<td>0.86</td>
<td>0.84</td>
<td>0.066</td>
<td>0.065</td>
</tr>
<tr>
<td>BMIMR</td>
<td>-1.923</td>
<td>6</td>
<td>0.92</td>
<td>0.44</td>
<td>0.48</td>
<td>0.073</td>
<td>0.080</td>
</tr>
</tbody>
</table>

Fig. 4. The optimal solution of Example 1

17 iterations were needed, using MR, to consider all the feasible MILP sub-problems. The number of iterations decreased to 13 as a results of using IMR instead (the non-considered graphs were excluded). The solution time of the NLP and MILP sub-problems, 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 bigger. (2) The number of equations containing binary variables also increased; therefore, the MILP subproblems became more complex. (3) The relaxation is weaker here because of using integer variables instead of binary ones. On the other hand, only 6 iterations were necessary in this case to investigate all the feasible MINLP sub-problems, thanks to having less number of binary variables. This is, perhaps, why the total solution time decreased.

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 et al. (2002). The flow rate of the feed to the membrane subsystem is 50 kg/hr; and it contains 10.05 mol % of water. The target is to reach 4 mol % of water in the final retentate. The system consists of maximum $k$ number of consecutive membrane sections, and in each section maximum the same $k$ number of membrane modules can be built in. 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. GAMS DICOPT++ with
CONOPT NLP-solver and OSL MILP-solver on a Sun Sparc station were used for
solving the MINLP representations. The initial value for each binary variable was 0.5.
Maximum 50 main iterations, and in all the iterations maximum 50000 iteration steps,
were allowed. The solution data are shown in Table 2.

Table 2. Summary of the computation results of Example 2

<table>
<thead>
<tr>
<th>MINLP repr.</th>
<th>k</th>
<th>cost ($/yr)</th>
<th>number of iterations</th>
<th>solution time NLP (s)</th>
<th>solution time MILP (s)</th>
<th>NLP/it. (s)</th>
<th>MILP/it. (s)</th>
<th>non-ideality</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMR</td>
<td>5</td>
<td>4619</td>
<td>7</td>
<td>8.42</td>
<td>0.39</td>
<td>8.03</td>
<td>0.056</td>
<td>1.147</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>4619</td>
<td>3</td>
<td>10.77</td>
<td>0.34</td>
<td>10.43</td>
<td>0.113</td>
<td>3.477</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>4619</td>
<td>3</td>
<td>39.65</td>
<td>0.47</td>
<td>39.18</td>
<td>0.157</td>
<td>13.060</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>NSXMR</td>
<td>8</td>
<td>4619</td>
<td>3</td>
<td>125.08</td>
<td>0.67</td>
<td>124.41</td>
<td>0.223</td>
<td>41.470</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>4619</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>613754</td>
</tr>
<tr>
<td>IMR</td>
<td>9</td>
<td>4619</td>
<td>4</td>
<td>4.08</td>
<td>0.76</td>
<td>3.32</td>
<td>0.190</td>
<td>0.830</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>4619</td>
<td>3</td>
<td>2.65</td>
<td>0.85</td>
<td>1.80</td>
<td>0.283</td>
<td>0.600</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BDIMR</td>
<td>11</td>
<td>4619</td>
<td>4</td>
<td>6.6</td>
<td>2.42</td>
<td>4.18</td>
<td>0.605</td>
<td>1.045</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>4619</td>
<td>4</td>
<td>7.84</td>
<td>2.69</td>
<td>5.15</td>
<td>0.673</td>
<td>1.288</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>4619</td>
<td>4</td>
<td>8.71</td>
<td>3.41</td>
<td>5.30</td>
<td>0.853</td>
<td>1.325</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>4619</td>
<td>3</td>
<td>6.88</td>
<td>4.37</td>
<td>2.51</td>
<td>1.457</td>
<td>0.837</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>4619</td>
<td>3</td>
<td>8.15</td>
<td>5.83</td>
<td>2.32</td>
<td>1.943</td>
<td>0.773</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>4619</td>
<td>4</td>
<td>13.28</td>
<td>6.63</td>
<td>6.65</td>
<td>1.658</td>
<td>1.663</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

First we generated and solved BMR with $k=5$, and then with increased $k$. For $k=8$, and
for greater problems, the solver does not find integer solution within the iteration limit.
Then the graphs of non-considered structures were excluded from the representation;
this is denoted by NSXMR (Non-considered Structures eXcluded MR). Non-ideality of
the representation is decreased in this way. Case $k=8$, which was infeasible by BMR,
was solved using NSXMR. For $k=9$, the solver does not find integer solution. The IMR
was then generated from NSXMR with additional constraints in order to exclude all the
non-considered isomorphic graphs from the representation. The case $k=9$, insolvable by
NSXMR, was solved using IMR, and then the size of the problem was increased. The
maximum size of problems solvable with IMR was the case $k=10$.

![Fig. 5. NLP solution time per iterations](image1)

![Fig. 6. MILP solution time per iterations](image2)

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. Further advance is expected, however, by decreasing the number of binary variables. We have only an approximation for the minimum number of binary variables because the number of represented graphs is not known exactly. A possible way to decrease the number of binary variables is using minimum number of binary variables in each section. By this way a Binarily Decreased and Ideal MR (BDIMR) is generated. Using this representation the maximum size of problems solvable is $k=16$.

In each case, the optimal solution contains 11 membrane modules. The NLP solution time per iterations (Fig. 5) increases monotonically 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 sub-problems. The MILP solution time per iterations increases exponentially with the problem size in case of BMR and NSXMR. Data for BMR are shown, but the point of NSXMR is not represented in Fig. 6 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 were 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 BMR at $k=5$ (including maximum 25 membrane modules).

6. Conclusion

Both assignment of considered graphs and formulation of the MINLP representation have significant effect on the solvability and solution time of process synthesis problems.

The solution time of a small synthesis problem is reduced in a high ratio when ideal and binarily minimal MINLP representation is used.

The solvable scale of an industrial membrane system synthesis problem, characterized by a high value of structural redundancy, increased from appr. $7^2=49$ to appr. $16^2=256$ potential membrane units as results of applying both idealization and reduction of the number of binary variables.

Acknowledgement: This study has been supported by OTKA T037191 and F035085.

References


