1 Introduction

In a manufacturing system, materials of different properties are consumed through various mechanical, physical and chemical transformations to result in desired products. Devices in which these transformations are carried out are called operating units, e.g., a lathe or a chemical reactor. Hence, a manufacturing system can be considered as a network of operating units which is called process network. The importance of process network synthesis (PNS) arises from the fact that such networks are ubiquitous in the chemical and allied industries. A process design problem in general, and flowsheeting in particular mean to construct a manufacturing system. A design problem is defined from a structural point of view by the raw materials, the desired products, and the available operating units, which determine the structure of the problem as a process graph containing the corresponding interconnections among the operating units. Thus, the appropriate process networks can be described by some subgraphs of the process graph belonging to the design problem under consideration. Naturally, the cost minimization of a process network is indeed essential. For this purpose, several papers have appeared for solving PNS problems by global optimization methods (cf. [9] and [18]) and by combinatorial approach based on the feasible graphs of processes (see, e.g., [17], [14], [10]). However its solution is difficult in general. It has been pointed out that the PNS-problems are NP-complete (see [1], [16] and [26]).

In general, there are three basic approaches to attack NP-hard problems. The first approach is to develop exponential time algorithms for solving the problem. In case of PNS problem, some exponential time algorithms based on the Branch and Bound technique were developed and studied in [15], [11], [12], [13], [23], [24], [19], [20], and [22]. This approach is studied also in this work. Another approach is to investigate specially structured instances for which interesting stuctural properties and bounds (e.g. for number of feasable solutions, see [2], [3], [4]) can be determined, which are called *well-solvable classes* and their instances can be solved efficiently. Some well-solvable classes were presented in [29], [5], [5], [30]. The third approach is to establish fast (polynomial time) algorithms which do not guarantee an optimal solution in general, but always result in a feasible solution which is close to the optimal solution in some sense. Such algorithms, called *heuristic algorithms* or *heuristics*, are important for several reasons. The feasible solutions found by such algorithms can be used in exponential time algorithms, furthermore, there is often not enough time to find an optimal solution or the size of the problem is too large to use an exponential algorithm. In these cases, heuristic algorithms can be useful again. The first heuristic algorithm for this problem was presented in [7].

2 The PNS problem

Let M be a given finite set of objects which are materials capable of being converted or transformed by a process. Transformation between two subsets of M occurs in an operating unit. It is necessary to link this operating unit to others through the elements of these two subsets of M. The resultant structure is a process graph (see [10]) defined as follows.

Let M be a finite nonempty set, and also let $O \subseteq \varphi'(M) \times \varphi'(M)$ with $O \neq \emptyset$ and $M \cap O = \emptyset$, where $\varphi'(M)$ denotes the set of all nonempty subsets of M. The elements of O are called *operating units*: for operating unit $u = (\alpha, \beta) \in O$, α and β are called the *input-set* and *output-set* of u, respectively. Pair (M, O) is defined as *process graph* or P-*graph* in short. The set of vertices of this graph is $M \cup O$, and the set of arcs is $A = A_1 \cup A_2$ with $A_1 = \{(X, Y) : Y = (\alpha, \beta) \in O \text{ and } X \in \alpha\}$ and $A_2 = \{(Y, X) : Y = (\alpha, \beta) \in O \text{ and } X \in \beta\}$. If there exist vertices $X_1, X_2, ..., X_n$, such that $(X_1, X_2), (X_2, X_3), \ldots, (X_{n-1}, X_n)$ are arcs of process graph (M, O), then $[X_1, X_n]$ is defined to be a *path* from vertex X_1 to vertex X_n. Let process graphs (m, o) and (M, O) be given; (m, o) is defined to be a *subgraph* of (M, O), if $m \subseteq M$, $o \subseteq O$ and $o \subseteq \varphi'(m) \times \varphi'(m)$.

To define a structural model of PNS, the set of materials to be included in the model need be specified. In the sequal, each material is an element of M, an arbitrarily specified finite set of the available materials. From the technical point of view, we suppose that $M \cap (\varphi'(M) \times \varphi'(M)) = \emptyset$. Now, a process design problem can be defined from a structural point of view in the following way. By a *structural model* of PNS, we mean the triplet, $\mathbf{M} = (P, R, O)$, where $P \subseteq M$ and $O \subseteq \varphi'(M) \times \varphi'(M)$ are finite nonempty sets representing the set of desired products and that of available operating units, respectively, and $R \subseteq M$ is a finite set representing the set of raw materials. Moreover, $P \cap R = \emptyset$, and α, β are finite sets for all operating units $u = (\alpha, \beta) \in O$.

Now, let $\mathbf{M} = (P, R, O)$ be a structural model of PNS; then, we can assign a P-graph to $\mathbf{M}$ as follows. Let M' denote the set of materials belonging to the operating units from O and M denote set $M' \cup P \cup R$. It can be seen that M and O are nonempty finite sets and that $O \subseteq \varphi'(M) \times \varphi'(M)$ and $M \cap O = \emptyset$. Thus, (M, O) is a P-graph representing the interconnections among the operating units in set O. Since $M \cap O = \emptyset$, the vertices which are the points in (M, O) can be divided into the two disjoint sets, M and O. The elements of M are called *material points* and those of O, *unit points* of (M, O). A subgraph of (M, O) can be assigned to each feasible process of the PNS problem; such a subgraph represents the structure or network of the process under consideration. If additional constraints, e.g., the material balance, are disregarded, the subgraphs of (M, O), which can be assigned to the feasible processes, have common combinatorial properties. Such properties, explored in [14], are given below.

Subgraph (m, o) of (M, O) is called a *feasible solution* of $\mathbf{M} = (P, R, O)$ if the following properties are satisfied.

($\mathcal{A}1$) $P \subseteq m$,

($\mathcal{A}2$) $\forall X \in m$, $X \in R \Leftrightarrow$ there exists no (Y, X) arc in (m, o),

($\mathcal{A}3$) $\forall Y_0 \in o$, $\exists$ path $[Y_o, Y_n]$ with $Y_n \in P$,

($\mathcal{A}4$) $\forall X \in m$, $\exists(\alpha, \beta) \in o$ such that $X \in \alpha \cup \beta$.

Let us denote the set of the feasible solutions of $\mathbf{M}$ by $S(\mathbf{M})$. It is easy to see that $S(\mathbf{M})$ is closed under the finite union. Consequently,

$$\cup\{(m, o) : (m, o) \in S(\mathbf{M})\}$$

is also a feasible solution provided that $S(\mathbf{M}) \neq \emptyset$; it is the greatest feasible solution with respect to the relation, subgraph ordering. This distinguished graph is called the *maximal structure* of $\mathbf{M}$.

Now, a simple class of PNS problems can be defined, a class of such PNS problems in which each operating unit has a positive fixed charge. We are to find a feasible process with the minimum cost. Each feasible process in this class of PNS problems is uniquely determined from the corresponding feasible solution and vice versa. Hence, the problem under consideration can be formalized as follows. Let a structural model of PNS problem $\mathbf{M} = (P, R, O)$ be given; moreover, let z be a positive real-valued function defined on S($\mathbf{M}$). The basic model is then

$$(\textit{PNS-1}) \qquad \min\{z((m, o)) \ : \ (m, o) \in S(\mathbf{M})\}.$$

Let $w : O \to R_+$ the fixed charge function defined for operating units. If by the cost of a process, we mean the sum of the fixed charges of the operating units belonging to the process of interest, we have to solve

$$(\textit{PNS-2}) \qquad \min\{\textstyle\sum_{u \in o} w(u) \ : \ (m, o) \in S(\mathbf{M})\}.$$

It has been proved that this PNS problem is NP-complete (see [1], [16] and [26]); therefore, the branch-and-bound technique may be an appropriate tool for its solution. In the branch-and-bound procedures for solving PNS problems, the notion of the decision-mapping (see [11]) has been introduced. Let $\mathbf{M} = (P, R, O)$ be a structural model of PNS. Then, P-graph (M, O) of $\mathbf{M}$ determines a function Δ of $M \setminus R$ into $\varphi'(O)$ as follows. For any material $X \in M \setminus R$, let

$$\Delta(X) = \{(\alpha, \beta) : (\alpha, \beta) \in O \ \& \ X \in \beta\}.$$

Let m be a subset of $M \setminus R$; furthermore, let $\delta(X)$ be a subset of $\Delta(X)$ for each $X \in m$. Mapping δ from set m into the set of subsets of O, $\delta[m] = \{(X, \delta(X)) : X \in m\}$, is called a *decision-mapping belonging* to $\mathbf{M}$; $\delta[m]$ is said to be *consistent* when $\delta(X) \cap \Delta(Y) \subseteq \delta(Y)$ is valid for all $X, Y \in m$, and the set of all consistent decision-mappings of $\mathbf{M}$ is denoted by $\Omega_{\mathbf{M}}$. In particular, if $\delta[m] \in \Omega_{\mathbf{M}}$ and $m = M \setminus R$, then sometimes we use the shorter notation δ instead of $\delta[M \setminus R]$. A decision-mapping can be visualised as a sequence of decisions, each of which is concerned with a single material involved in the process being synthesized; it identifies the set of operating units to be considered for producing directly the material of interest. The meaning of the consistency can be presented as follows. Material X is to be produced by operating units included in $\delta(X)$. Then, those opertaing units of $\delta(X)$ that also participate in the production of material Y, i.e., $\delta(X) \cap \Delta(Y)$, must be considered for the production of material Y, and thus, $\delta(Y) \supseteq \delta(X) \cap \Delta(Y)$.

We define function *op* on $\Omega_{\mathbf{M}}$ for selecting the set of those operating units that are decided to produce any of the materials in set m based on consistent decision-mapping $\delta[m]$. Formally, for any $\delta[m] \in \Omega_{\mathbf{M}}$,

$$op(\delta[m]) = \cup\{\delta(X) : X \in m\}.$$

Furthermore, we need the following functions. For any finite set of operating units o, let

$$mat^{in}(o) = \cup_{(\alpha,\beta)\in o}\alpha, \qquad mat^{out}(o) = \cup_{(\alpha,\beta)\in o}\beta.$$

Let $\delta_1[m_1]$ and $\delta_2[m_2]$ be arbitrary consistent decision-mappings. Then, $\delta_2[m_2]$ is called an *extension* of $\delta_1[m_1]$ if $m_1 \subseteq m_2$ and $\delta_1(X) = \delta_2(X)$ for all $X \in m_1$; this is denoted by $\delta_1[m_1] \leq \delta_2[m_2]$. In particular, if $\delta_1[m_1] \leq \delta_2[m_2]$ and $m_1 \subset m_2$, a *proper extension* exists; it is denoted by $\delta_1[m_1] < \delta_2[m_2]$. Relation extension is reflexive, antisymmetric and transitive; hence, it is a partial ordering on $\Omega_{\mathbf{M}}$. Let us denote the set of all maximal elements of this partially ordered set by $\Omega_{\mathbf{M}}^{\max}$.

3 The number of the consistent decision-mappings

Thesis 1.

Theorem 3.1. ([2]) *For every $\emptyset \neq m \subseteq M \setminus R$, the number of the all decision-mappings defined on m is $2^{\sum_{X\in m}|\Delta(X)|}$.*

Let us denote by $\tau(m)$ the number of the consistent decision-mappings defined on m.

Theorem 3.2. ([2]) *For every $\emptyset \neq m \subseteq M \setminus R$, $\tau(m) = 2^{|\cup\{\Delta(X):X\in m\}|}$.*

Remark 1. In particular, if $m = M \setminus R$, then $\tau(m) = 2^{|O|}$. This shows that there is a strong relationship between the maximal consistent decision-mappings and the subsets of O. Indeed, it can be proved that mapping γ defined by $\gamma(\delta) = op(\delta)$ is a one-to-one mapping of $\Omega_{\mathbf{M}}^{\max}$ onto $\varphi(O)$ where $\varphi(O)$ denotes the set of all subsets of O.

Regarding the relationship between the maximal decision-mappings and the feasible solutions, let us define mapping ρ in the following way. For any $(m, o) \in S(\mathbf{M})$, let $\rho(m, o) = \delta$ where δ is defined by

$$\delta(X) = \{u : u = (\alpha, \beta) \in o \,\&\, X \in \beta\}$$

for all $X \in M \setminus R$. It can be easily proved that ρ is a one-to-one mapping of $S(\mathbf{M})$ into $\Omega^{\max}_{\mathbf{M}}$. Therefore, $2^{|O|}$ is a trivial upper bound for $|S(\mathbf{M})|$. Taking into account property (A2), this bound can be improved as follows.

Let $(m, o) \in S(\mathbf{M})$ be an arbitrary feasible solution and $\rho(m, o) = \delta$. Then, (A2) implies the following inclusion:

$$(\mathcal{A}'2) \qquad mat^{in}(op(\delta)) \subseteq mat^{out}(op(\delta)) \cup R.$$

Let us denote by $\tau'(m)$ the number of the consistent decision-mappings defined on m satisfying $(\mathcal{A}'2)$. Then $\tau'(m) \geq |S(\mathbf{M})|$.

Let $O = \{u_1, \ldots, u_n\}$, $M = \{X_1, \ldots, X_k\}$, $O(X_j) = \{u : u = (\alpha, \beta) \in O \;\&\; X_j \in \alpha\}$, for all $X_j \in M$, and for all $j \in \{1, \ldots, k\}$

$$A_j = \{\delta : \delta \in \Omega^{\max}_{\mathbf{M}} \;\&\; X_j \in mat^{in}(op(\delta)) \setminus (mat^{out}(op(\delta)) \cup R)\}$$

Then, $(\mathcal{A}'2)$ is not satisfied by δ and the reason is that $X_j \in mat^{in}(op(\delta))$ and $X_j \notin mat^{out}(op(\delta)) \cup R$. For every $\emptyset \neq I \subseteq \{1, \ldots, k\}$, let us define set A_I by $A_I = \cap_{i \in I} A_i$, and in particular, let $A_\emptyset = \Omega^{\max}_{\mathbf{M}}$. If $I = \{i_1, \ldots, i_l\}$, then

$$A_I = \left\{\delta : \delta \in \Omega^{\max}_{\mathbf{M}} \;\&\; \{X_{i_1}, \ldots, X_{i_l}\} \subseteq mat^{in}(op(\delta)) \setminus (mat^{out}(op(\delta)) \cup R)\right\}$$

Theorem 3.3. ([2])

$$\tau'(m) = |\Omega^{\max}_{\mathbf{M}} \setminus (A_1 \cup A_2 \cup \ldots \cup A_k)| = \Sigma_{I \subseteq \{1, \ldots k\}} (-1)^{|I|} \cdot |A_I| \,.$$

Remark 2. It is worth noting that the bound presented above is independent of the set of the required products. It is valid under arbitrary $P \subseteq M \setminus R$.

Unfortunately, to count $|A_I|$ is a difficult problem. In general case, we have to cover $\{X_{i_1}, \ldots, X_{i_l}\}$ with such a system, $\alpha_{j_1}, \ldots, \alpha_{j_s}$ for which there are operating units $(\alpha_{j_t}, \beta_{j_t}) \in O$, $t = 1, \ldots, s$, with $\{X_{i_1}, \ldots, X_{i_l}\} \cap \beta_{j_t} = \emptyset$, $t = 1, \ldots, s$, and $|A_I|$ is equal to the number of the such covering systems. The determination of $|A_I|$ is easier if we restrict ourselves to special classes of PNS problems. An interesting special case is the class containing separator type operating units, i.e., $|\alpha| = 1$ is valid for all $u = (\alpha, \beta) \in O$. Let us consider set $I = \{i_1, \ldots, i_l\}$ again. Let $O^*(X_{i_j}) = O(X_{i_j}) \setminus (\cup_{i \in I} \Delta\,(X_i))$. Then, $O^*(X_{i_j})$ is the set of operating units such that they do not produce any material from $\{X_t : t \in I\}$ and each of them has X_{i_j} as input material.

Theorem 3.4. ([2]) For separator type operating units

$$|A_I| = \left(\prod_{t=1}^{l} \left(2^{|O^*(X_{i_t})|} - 1\right)\right) \cdot 2^{|O \setminus (\cup_{i \in I} \Delta(X_i)) \setminus (\cup_{i \in I} O(X_i))|}$$

A model for PNS problem with separator type operating units is called *Line model*, if

$$u_1 = (\alpha_1, \beta_1) \text{ with } \alpha_1 = X_1 \text{ and } \beta_1 = X_2,$$
$$u_k = (\alpha_k, \beta_k) \text{ with } \alpha_k = X_k \text{ and } \beta_k = X_{k-1},$$

and in general:

$$u_i = (\alpha_i, \beta_i) \text{ with } \alpha_i = X_i \text{ and } \beta_i = \{X_{i-1}, X_{i+1}\}, (2 \le i \le k-1).$$

If in Line model we modify β_1 and β_k such that:

$$\beta_1 = \{X_2, X_k\} \text{ and } \beta_k = \{X_{k-1}, X_1\}$$

then we obtain an another model called *Chain model*.

Theorem 3.5. ([3]) In the Line model $|S(\mathbf{M})| \le L^{(1)}$ where

$$\begin{aligned} L^{(1)} = 2^k + \sum_{1\le j\le \frac{k+1}{2}} (-1)^j \cdot \Bigg[& \sum_{\substack{0\le r\le j-1\\ k-3j+r+2\ge 0}} \binom{j-1}{r} \cdot \binom{k-2j}{j-r-2} \cdot 2^{k-3j+r+2} + \\ & + \sum_{\substack{0\le r\le j-1\\ k-3j+r+1\ge 0}} \binom{j-1}{r} \cdot \binom{k-2j}{j-r-1} \cdot 2^{k-3j+r+1} + \\ & + \sum_{\substack{0\le r\le j-1\\ k-3j+r\ge 0}} \binom{j-1}{r} \cdot \binom{k-2j}{j-r} \cdot 2^{k-3j+r} \Bigg]. \\ = 1 + \sum_{2\le t\le k} \sum_{1\le q\le \min\{\frac{t}{2};k-t+1\}} & \binom{t-q-1}{q-1} \cdot \binom{k-t+1}{q}. \end{aligned}$$

Theorem 3.6. ([3]) In the Chain model $|S(\mathbf{M})| \le C^{(1)}$ where

$$C^{(1)} = 2^k + \sum_{1\le j<\frac{k}{2}} (-1)^j \cdot \sum_{\substack{0\le r\le j-1\\ k-3j+r\ge 0}} \frac{k}{j} \cdot \binom{j}{r} \cdot \binom{k-2j-1}{j-r-1} \cdot 2^{k-3j+r} + e_k =$$

$$\begin{aligned} = 1 + \sum_{2\le t\le k} \Bigg[& \sum_{1\le q\le \min\{\frac{t}{2};k-t\}} \binom{t-q-1}{q-1} \cdot \binom{k-t-1}{q-1} + \\ & + \sum_{2\le i\le t} \sum_{1\le q\le \frac{t-i}{2}+1} \binom{t-i-q+1}{q-1} \cdot \binom{k-t-1}{q-1} + \\ & + \sum_{1\le i\le k-t} \sum_{1\le q\le \min\{\frac{t}{2};k-t-i+1\}} \binom{t-q-1}{q-1} \cdot \binom{k-t-i}{q-1} \Bigg] + 1, \end{aligned}$$

where

$$e_k = \begin{cases} (-1)^{\frac{k}{2}} \cdot 2 & \text{, if } k \text{ even,} \\ 0 & \text{, if } k \text{ odd.} \end{cases}$$

4 Merging reduction

Thesis 2

Two operating units $u_1, u_2 \in O$ are called *mergeable* if $u_1 \in (m,o)$ implies $u_2 \in (m,o)$, and conversely, for every feasible solution $(m,o) \in S(\mathbf{M})$. This is an equivalence relation on set O which is denoted by $\equiv$. Let us define structural model $\mathbf{M}/\equiv = (P, R, O^*)$ by

$$O^* = \{(\cup\{\alpha_t : u_t = (\alpha_t, \beta_t) \in C(u)\}, \cup\{\beta_t : u_t = (\alpha_t, \beta_t) \in C(u)\}) : u \in O\}$$

where $C(u)$ denotes the equivalence classes containing u. Let us define a mapping $\Psi : M \cup O \longrightarrow M \cup O^*$ by

$$\begin{cases} \Psi(X) = X & \text{, if } X \in M, \\ \Psi(u_s) = (\cup\{\alpha_t : u_t \in C(u)\}, \cup\{\beta_t : u_t \in C(u)\}) & \text{, if } u_s \in C(u), \\ \Psi(m) = \{\Psi(X) : X \in m\} & \text{, if } m \subseteq M, \text{ and} \\ \Psi(o) = \{\Psi(u) : u \in o\} & \text{, if } o \subseteq O \end{cases}$$

Theorem 4.1. ([19]) *Mapping Ψ is a bijective mapping of $S(\mathbf{M})$ onto $S(\mathbf{M}/\equiv)$.*

For any $(m,o) \in S(\mathbf{M})$ and $(m^*, o^*) \in S(\mathbf{M}/\equiv)$, let $w(m,o) = \sum\{w(u) : u \in o\}$ and $\bar{w}(m^*, o^*) = \sum\{w(u) : u \in o^*\}$. Then, $w(m,o) = \bar{w}(\Psi(m,o))$ is valid, for all feasible solution $(m,o) \in S(\mathbf{M})$. Let

$$\text{(\textit{PNS-6})} \qquad \min\left\{\sum_{u \in o} \bar{w}(u) : (m,o) \in S(\mathbf{M}/\equiv)\right\}.$$

Theorem 4.2. ([19]) *The image of an optimal solution of problem PNS-2 under Ψ is an optimal solution of problem PNS-6, and conversely, the image of an optimal solution of problem PNS-6 under Ψ^{-1} is an optimal solution of problem PNS-2.*

Let $\mathbf{M} = (P, R, O)$ be a reduced structural model of PNS with $S(\mathbf{M}) \neq \emptyset$. Furthermore, let $u_j \in O$ be arbitrary. Then, we can construct a new structural model of PNS $\mathbf{M}(u_j) = (P, R, O \setminus \{u_j\})$. Let us denote the maximal structure of $\mathbf{M}(u_j)$ by (M_j, O_j) provided that it exists. If it does not exist, then let $M_j = O_j = \emptyset$.

Theorem 4.3. ([19]) *For every $u_i, u_j \in O$, $u_i \equiv u_j$ if and only if $u_i \in O \setminus O_j$ and $u_j \in O \setminus O_i$ are simultaneously valid.*

Then we obtain the following procedure to determine the required equivalence relation where it is assumed that $O = \{u_1, \ldots, u_n\}$.

Procedure ER for determination of equivalence relation ([19])

- *Step 1.* Set $i := 1$, $k := 1$, $N = \{1, \ldots, n\}$.

- *Step 2.* Determine the maximal structure of $\mathbf{M}(u_i)$ by the maximal structure generation algorithm. If $O_i = O \setminus \{u_i\}$, then let $V_k = \{u_i\}$, $N = N \setminus \{i\}$, $k = k+1$, and proceed to Step 3. Otherwise, proceed to Step 3.

- *Step 3.* If $i = n$, then proceed to Step 4. Otherwise, let $i = i+1$, and proceed to Step 2.

- *Step 4.* Terminate if $N = \emptyset$. Otherwise, let i denote the smallest element of N. Let $J = \{t : t \in N\ u_t \in O \setminus O_i\}$. Let $V = \emptyset$, and proceed to Step 5.

- *Step 5.* If $J = \emptyset$, then let $N = N \setminus \{i\}$, $V_k = V \cup \{u_i\}$, $k = k+1$, and proceed to Step 4. Otherwise, proceed to Step 6.

- *Step 6.* Choose an element j from J. Let $J = J \setminus \{j\}$. If $u_i \in O \setminus O_j$, then let $V = V \cup \{u_j\}$, $N = N \setminus \{j\}$, and proceed to Step 5. Otherwise, proceed to Step 5.

As a result of this procedure, we obtain the equivalence classes belonging to the required equivalence relation as $V_1, \ldots, V_k$.

5 Look Ahead B&B Algorithm

Thesis 3

For every $(m, o) \in S(\mathbf{M})$, let us assign decision mapping $\rho((m, o)) = \delta[M \setminus R]$ to (m, o) according to

$$\delta(X) = \{(\alpha, \beta) : (\alpha, \beta) \in o \text{ and } X \in \beta\} \text{ if } X \in m \setminus R, \text{ and}$$

$$\delta(X) = \emptyset \text{ if } X \in M \setminus (R \cup m).$$

Let $S'(\mathbf{M}) = \{\rho((m, o)) : (m, o) \in S(\mathbf{M})\}$. Then it is possible to solve the problem stated below instead of problem PNS-2:

$$(PNS\text{-}5) \qquad \min \left\{ \sum_{u \in op(\delta)} w(u) : \delta \in S'(\mathbf{M}) \right\}.$$

Let $\delta[m] \in \Omega_{\mathbf{M}}$ with $|m| < |M \setminus R|$; moreover, let $O_{\delta[m]} = op(\delta[m]) \cup (\bigcup\{C(u) : u \in op(\delta[m])\})$. Let $Y \in (mat^{in}(O_{\delta[m]}) \cup P) \setminus (mat^{out}(O_{\delta[m]}) \cup R)$ provided that this set is not empty. Let us denote by $K_1, \ldots, K_r$ the equivalence classes of $\Delta(Y)$ with respect to the restriction of equivalence relation $\equiv$ to $\Delta(Y)$. For every nonempty subset J of $\{K_1, \ldots, K_r\}$, let $K_J = \cup\{K_t : K_t \in J\}$. Then, a consistent decision-mapping of the form, $\delta_t[m \cup \{Y\}] = \delta[m] \cup \{(Y, K_J)\}$, $J \subseteq \varphi'(\{K_1, \ldots, K_r\})$, is called *irregular extension of* $\delta[m]$ *with respect to* Y if $O_{\delta(A)} \cap \Delta(B) \subseteq \delta(B), \forall A, B \in m \cup \{Y\}$. Obviously, every irregular extension is an extension as well. Let us consider the reflexive, transitive closure of the relation, irregular extension, on $\Omega_{\mathbf{M}}$. Obviously, the resultant relation is a partial ordering denoted by $\preceq$. Let $\Sigma_{\mathbf{M}}$ is defined by

$$\Sigma_{\mathbf{M}} = \{\delta[m] : \delta[m] \in \Omega_{\mathbf{M}} \ \& \ \delta_0[\emptyset] \preceq \delta[m]\}$$

and its elements are called *irregular decision-mappings*, where $\delta_0[\emptyset] = \emptyset$. Let

$$S^*(\mathbf{M}) = \{\mathrm{icl}(\delta[m]) : \delta[m] \in \Sigma_{\mathbf{M}} \,\&\, (mat^{in}(O_{\delta[m]}) \cup P) \setminus (mat^{out}(O_{\delta[m]}) \cup R) = \emptyset\}.$$

Proposition 5.1. ([20]) *$S^*(\mathbf{M}) \subseteq S'(\mathbf{M})$.*

Proposition 5.2. ([20]) *If (m, o) is an optimal solution of problem PNS-2, then $\rho((m, o))$ is contained in $S^*(\mathbf{M})$.*

Theorem 5.1. ([20]) It is possible to solve the problem below instead of problem PNS-5.

$$(PNS\text{-}7) \qquad \min\left\{\sum_{u \in op(\delta)} w(u) : \delta \in S^*(\mathbf{M})\right\}.$$

Look Ahead B&B Algorithm ([20])

Initialization

- Compute the merging equivalence relation.
- Let $L := \{\varphi(\delta_0[\emptyset])\}$, $z^* := \infty$, $s := \emptyset$, and $r := 0$. Compute $g^*(\delta_0[\emptyset])$.

Iteration (r-th iteration)

- *Step 1.* (*Termination*) Terminate if $L = \emptyset$: set s contains the optimal solution and z^* provides the optimal value. Otherwise, proceed to Step 2.
- *Step 2. (Selection)* Choose its element if L is a singleton; otherwise choose an element, $\varphi(\delta[m])$, of L for which ratio $g^*(\delta[m])/|m|$ is minimal. (If there are more than one candidate with the same ratio, then choose one of them randomly.) Proceed to Step 3.
- *Step 3. (Testing)* If $T = (mat^{in}(O_{\delta[m]}) \cup P) \setminus (mat^{out}(O_{\delta[m]}) \cup R) \neq \emptyset$, then proceed to Step 4. Otherwise, form the irregular closure δ' of $\delta[m]$ and redefine the value of z^* and set s by $z^* := w(\delta')$ and $s := \{\delta'\}$, respectively, if $w(\delta') < z^*$. If not, then z^* and s do not change. Let $\Phi := \emptyset$ and proceed to Step 6.

- *Step 4. (Branching)* Choose an X from T for which $|(mat^{out}(\Delta(X)) \cap T|$ is maximal, and form the irregular extensions of $\delta[m]$ with respect to X. Let $L := L \setminus \{\varphi(\delta[m])\}$ if there is no such irregular extension of $\delta[m]$ and proceed to Step 1. Otherwise, denote the irregular extensions of $\delta[m]$ with respect to X by $\delta_i[m_i]$, $i = 1, 2, \ldots, k$. Then, let $\Phi = \{\varphi(\delta_i[m_i]) : 1 \leq i \leq k\}$, and proceed to Step 5.

- *Step 5. (Bounding)* Calculate the value of $g^*(\delta_i[m_i])$, $i = 1, 2, \ldots, k$, and proceed to Step 6.

- *Step 6. (Fathoming)* Redefine set L as follows:

$$L := \{\varphi(\bar{\delta}[\bar{m}]) : \varphi(\bar{\delta}[\bar{m}]) \in (L \setminus \{\varphi(\delta[m])\}) \cup \Phi,\ g^*(\bar{\delta}[\bar{m}]) < z^*\}.$$

  Set $r := r + 1$ and iterate.

6 Partial Enumeration Algorithm

Thesis 4.

The following procedure does not enumerate all of the feasible solutions in general, but it provides all of the optimal solutions. Its advantage is that it needs fewer decision-mappings than the complete enumeration presented in [15].

Partial Enumeration ([22])

Initialization

- Compute the merging equivalence relation and let us denote by o_0 the set of operating units which must be in every solution structure. Set $m_0 := \emptyset$ and $i := 0$.

Iteration

- *Step 1.* Let $\delta_i[m_i]$ be the current irregular decision-mapping where the ordered domain is $\hat{m}_i =< A_{j_1}, \ldots, A_{j_k} >$. Furthermore, let

$$T_i = (mat^{in}(O_{\delta_i[m_i]}) \cup P) \setminus (mat^{out}(O_{\delta_i[m_i]}) \cup R),$$

  and proceed to Step 2.

- *Step 2.* Construct the irregular closure of $\delta_i[m_i]$ if $T_i = \emptyset$; denote it by δ_i'; redefine S by $S := S \cup \{\delta_i'\}$, and proceed to Step 4. Otherwise, proceed to Step 3.

- *Step 3.* Select the material denoted by X from T_i with the smallest index for which $|mat^{out}(\Delta(X)) \cap T_i|$ is maximal. Investigate the irregular extensions of $\delta_i[m_i]$ for the corresponding subsets of $\Delta(X) \setminus o_0$ with respect to the linear ordering introduced above.

  Choose the first subset, K_J, of $\Delta(X) \setminus o_0$, for which $\delta_i[m_i] \cup \{(X, K'_J)\}$ is an irregular extension of $\delta_i[m_i]$, where

$$K'_J = K_J \cup (\Delta(X) \cap o_0)$$

  provided that such a K'_J there exists. Let $\hat{m}_{i+1} = \hat{m}_i \bigvee \{X\}$, and

$$\delta_{i+1}[m_{i+1}] = \delta_i[m_i] \cup \{(X, K'_J)\},$$

  Set $i := i + 1$, and proceed to the succeeding iteration step.

  If none of the corresponding subsets of $\Delta(X)$ yields an irregular exten- sion of $\delta_i[m_i]$, then proceed to Step 4.

- *Step 4.* If $\delta_i(A_{j_k}) \subset \Delta(A_{j_k})$ and for some subset K_J of $\Delta(A_{j_k}) \setminus o_0$ with $\delta_i(A_{j_k}) \neq K'_J$ and $\delta_i(A_{j_k}) \leq K'_J$, where

$$K'_J = K_j \cup (\Delta(X) \cap o_0)$$

  decision-mapping

$$\{(A_{j_1}, \delta_i(A_{j_1})\} \cup \cdots \cup \{(A_{j_{k-1}}, \delta_i(A_{j_{k-1}})\} \cup \{(A_{j_k}, K'_J)\}$$

  is an irregular extension of $\{(A_{j_1}, \delta_i(A_{j_1})\} \cup \cdots \cup \{(A_{j_{k-1}}, \delta_i(A_{j_{k-1}})\}$, then proceed to Step 5. Otherwise, proceed to Step 6.

- *Step 5.* Choose the first subset of $\Delta(A_{j_k}) \setminus o_0$ with respect to the linear ordering of the corresponding subsets of $\Delta(A_{j_k}) \setminus o_0$ which satisfies the condition of Step 4; denote it by K_J. Define $\hat{m}_{i+1}$ and $\delta_{i+1}[m_{i+1}]$ as follows:

$$\hat{m}_{i+1} =< A_{j_1}, \ldots, A_{j_k} >,$$

$$\delta_{i+1}[m_{i+1}] = \{(A_{j_1}, \delta_i(A_{j_1})\} \cup \cdots \cup \{(A_{j_{k-1}}, \delta_i(A_{j_{k-1}})\}) \cup \{(A_{j_k}, K'_J)\},$$

  where

$$K'_J = K_j \cup (\Delta(X) \cap o_0)$$

  Set $i := i + 1$, and proceed to the succeeding iteration step.

- *Step 6.* Redefine the value of k by $k := k - 1$. Terminate, if $k = 0$. Otherwise, proceed to Step 4.

Regarding the correctness of the procedure above, it can be easily seen that S contains the elements of $S^*(\mathbf{M})$ exactly. Consequently, S contains all of the optimal solutions of PNS-5.

7 Automaton Theory Approach

We use a coloring of process graphs. For this purpose, let $(\bar{M}, \bar{O})$ be a process graph and R a set of materials. It is said that $(\bar{M}, \bar{O})$ is *colorable* by R if every material vertex of $(\bar{M}, \bar{O})$ can be colored by the following procedure.

Coloring Procedure ([27])

Step 1. Color every material in $\bar{M} \cap R$.

Step 2. If there is an operating unit whose all input materials have already colored, then color its every output material. Terminate otherwise.

A *modified PNS problem* is a PNS problem satisfying $(\mathcal{A}1)$ - $(\mathcal{A}4)$ and

$(\mathcal{A}5)$ $(\bar{M}, \bar{O})$ is colorable by R.

The weighted modified PNS problem is then:

$$(PNS\text{-}8) \qquad \min \left\{ \sum_{u \in \bar{O}} w(u) : (\bar{M}, \bar{O}) \in \overline{S}(\mathbf{M}) \right\}.$$

where $\overline{S}(\mathbf{M})$ is the set of feasible solutions of the modified PNS problem.

Let $\mathbf{B} = (B, O')$ be an automaton defined as follows. Let $B = B' \cup \{\diamond\}$ with $B' = \varphi'(M)$ and $\diamond \notin B'$. Moreover, let $O' = \{u : u = (C, D) \in O \text{ and } R \cap D = \emptyset\}$. For every $Q \in B'$ and $u = (C, D) \in O'$, let

$$Qu^{\mathbf{B}} = \begin{cases} Q \cup D & \text{if } C \subseteq Q \\ \diamond & \text{otherwise,} \end{cases}$$

moreover

$$\diamond u^{\mathbf{B}} = \diamond.$$

Let us define now the recognizer $\mathcal{B} = (\mathbf{B}, R, F)$ by $F = \{Q : Q \in B' \text{ and } P \subseteq Q\}$.

For any $u, v \in O'$ let $v \ll u$ if $u = v$ or $mat^{out}(v) \cap mat^{in}(u) \neq \emptyset$. Let us denote by $\overset{*}{\ll}$ the transitive closure of $\ll$. It is said that two operating units $u, v \in O'$ are *mutually reachable* if $u \overset{*}{\ll} v$ and $v \overset{*}{\ll} u$. It is an equivalence relation on O' which will be denoted by $\bowtie$. Let $\mathcal{C} = O'/\bowtie$. For any $C, C' \in O'/\bowtie$ let $C \lll C'$ if $C = C'$ or there are $u \in C'$ and $v \in C$ such that $v \overset{*}{\ll} u$. hen $\lll$ is a partial ordering wich can be extended to a linear ordering. Let us define the function $f : (O')^* \to \{1, \ldots, h\}$ as follows. Let $f(\lambda) = 0$. For every $u \in O'$, let $f(u) = m$ if $u \in C_m$, moreover, for every word $p = u_1 \ldots u_l \in (O')^+$, let $f(p) = \max\{f(u_t) : 1 \leq t \leq l\}$. In what follows, by an *extended state* we mean a triplet $(Rp, p, w(p))$, where p is a word bringing the automaton from R into a state of F. We say that $(Rp, p, w(p))$ is an *optimal extended state* if it is an extended state and $w(p) \leq w(p')$ for any extended state $(Rp', p', w(p'))$.

Procedure PAT for finding an optimal solution of PNS-8 ([21])

Initialization. $i := 0,\ L_0 := \{(R, \lambda, 0)\}$.

Iteration

Step 1. $M_i := \{(Rp, p, w(p)) \in L_i : w(p) \leq w(q) \text{ for all } (Rq, q, w(q)) \in L_i\}$.
$S_i := \{(Rp, p, w(p)) \in M_i : P \subseteq Rp\}$.
If $S_i \neq \emptyset$, then terminate; the elements of S_i are optimal extended states.

Step 2. Select an arbitrary element $(Rt, t, w(t))$ from M_i with $t = u_1 \ldots u_n$.

Step 3. Let $i := i + 1$, $L_i := L_{i-1}$.

Step 4. $L_i := L_i \setminus \{(Rt, t, w(t))\}$.

Step 5. Let $V_i := \{v \in O' \setminus \{u_1, \ldots, u_n\} : mat^{in}(v) \subseteq Rt \text{ and } f(v) \geq f(t)\}$. If $V_i = \emptyset$, then proceed to Step 1. Otherwise, let $V_i = \{v_1, \ldots, v_m\}$.

Step 6. For every $j := 1$ to m do:
$A(v_j) := \{(Rq, q, w(q)) \in L_i : Rq \supseteq Rtv_j \text{ and } w(q) \leq w(tv_j) \text{ and } (w(q) < w(tv_j) \text{ or } f(q) \leq f(tv_j))\}$,
$D(v_j) := \{(Rq, q, w(q)) \in L_i : Rq \subseteq Rtv_j \text{ and } w(q) \geq w(tv_j) \text{ and } (w(q) > w(tv_j) \text{ or } f(q) \geq f(tv_j))\}$,
if $A(v_j) = \emptyset$, then let $L_i := (L_i \setminus D(v_j)) \cup \{(Rtv_j, tv_j, w(tv_j))\}$.

Step 7. Go to *Step 1.*

Theorem. ([20]) *The procedure PAT terminates in a finite number of steps and it provides an optimal extended state.*

References

[1] Blázsik, Z., B. Imreh, A note on connection between PNS and set covering problems, *Acta Cybernetica*, **12**, 1996, 309-312.

[2] Blázsik, Z., Cs. Holló, B. Imreh, On Decision-Mappings Related to Process Network Synthesis Problem, *Acta Cybernetica* **13**, 1998, 319-328.

[3] Blázsik, Z., Cs. Holló, B. Imreh, Explicit bound for the number of feasible solutions of special PNS-problem classes, *PU.M.A*, **9**, 1998, 17-27.

[4] Blázsik, Z., Cs. Holló, B. Imreh, Kiszámolható korlátok speciális PNS-problémaosztályok lehetséges megoldásai számára, *Új utak a magyar operációkutatásban, szerk. Komlósi, S., Szántai T., Dialóg Campus Kiadó, Budapest-Pécs,* 1999, 182-194.

[5] Blázsik, Z., Cs. Holló, B. Imreh, Cs. Imreh, Z. Kovács, On Bottleneck and k-sum version of the Process Network Synthesis Problem, *Novi Sad Journal Of Mathematics*, **3**, 2000, 11-19.

[6] Blázsik, Z., Cs. Holló, B. Imreh, Cs. Imreh, Z. Kovács, On a well-solvable class of the PNS problem, *Novi Sad Journal Of Mathematics*, **3**, 2000, 21-30.

[7] Blázsik, Z., Cs. Holló, Cs. Imreh, Z. Kovács, Heuristics for the Process Network Synthesis Problem, *New Trends in Equilibrium Systems, Mátraháza Optimization Days*, Kluwer Academic Publishers, 2000, 1-16.

[8] C. Croitoru, *Tehnici de bază în Optimizarea Combinatorie*, Ed. Universităţii Al. I. Cuza, Iaşi, 1992

[9] Floudas, C. A., I. E. Grossmann, Algorithmic Approaches to Process Synthesis: Logic and Global Optimization, AiChE Symposium Series No. 304, **91** (Eds: L. T. Biegler and M. F. Doherly), (1995), 198-221. .

[10] Friedler, F., K. Tarján, Y. W. Huang, and L. T. Fan, Graph-Theoretic Approach to Process Synthesis: Polynomial Algorithm for maximal structure generation, *Computer chem. Engng.* **17**, 1993, 924-942.

[11] Friedler, F., J. B. Varga, and L. T. Fan, Decision-Mappings: A Tool for Consistent and Complete Decisions in Process Synthesis, *Chem. Eng. Sci.*, **50** (11), 1995, 1755-1768.

[12] Friedler, F., J.B. Varga, E. Fehér, and L.T. Fan, Combinatorially Accerelated Branch-and-Bound Method for Solving the MIP Model of Process Network Synthesis, International Conference on State of the Art in Global Optimization: Computational Methods and Applications, Princeton, 1995.

[13] Friedler, F., J. B. Varga, E. Fehér, L. T. Fan, Combinatorially Accelerated Branch-and -Bound Method for Solving the MIP Model of Process Network Synthesis, *Nonconvex Optimization and its Applications*, (eds.: C. A. Floudas and P. M. Pardalos), Kluwer Academic Publishers, Norwell, MA, U.S.A., 1996, 609-626.

[14] Friedler, F., K. Tarján, Y. W. Huang, L. T. Fan, Graph-Theoretic Approach to Process Synthesis: Axioms and Theorems, *Chem. Eng. Sci.*, **47** (8), 1992, 1973-1988.

[15] Friedler, F., K. Tarján, Y. W. Huang, L. T. Fan, Combinatorial Algorithms for Process Synthesis, *Computer chem. Engng.*, **16**, 1992, 313-320.

[16] Friedler, F., J. Fulop, B. Imreh, On the reformulation of some classes of PNS-problems as set covering problems, *Acta Cybernetica*, **13**, 1998, 329-337.

[17] Friedler, F., L. T. Fan, B. Imreh, Process Network Synthesis: Problem Definition, *Networks*, **28**, 1998, 119-124.

[18] Grossmann, I. E., V. T. Voudouris, O. Ghattas, Mixed-Integer Linear Programming Reformulations for Some Nonlinear Discrete Design Optimization Problems, In: Recent Advances in Global Optimization (Eds: C. A. Floudas and P. M. Pardalos) Princeton University Press, New Jersey, 1992.

[19] Holló, Cs., Z. Blázsik, Cs. Imreh, Z. Kovács, On a Merging Reduction of the Process Network Synthesis Problem, *Acta Cybernetica* **14**, 1999, 251-261.

[20] Holló, Cs., A Look Ahead Branch-and-Bound Procedure for Solving PNS Problems, *PU.M.A.*, **2**, 2000, 265-279.

[21] Holló, Cs., A Procedure Based on Automaton Theory Approach for Solving Modified PNS Problems, *PU.M.A.*, **1-2**, 2002, 159-169.

[22] Holló, Cs., A Partial Enumeration Algorithm for Solving PNS Problems, *Mathematical and Computer Modelling*, **7-9**, 2003, 855-864.

[23] Imreh, B., F. Friedler, L. T. Fan, An Algorithm for Improving the Bounding Procedure in Solving Process Network Synthesis by a Branch-and-Bound Method, *Developments in Global Optimization*, ed. I. M. Bonze, T. Csendes, R. Horst, P. M. Pardalos, Kluwer Academic Publisher, Dordrecht, Boston, London, 1996, 301-348.

[24] Imreh, B., G. Magyar, Empirical Analysis of Some Procedures for Solving Process Network Synthesis Problem, *Journal of Computing and Information Technology*, **6**, 1998, 373-382.

[25] Imreh, B., *Kombinatorikus optimalizálás*, Novadat, 2000.

[26] Imreh, B., J. Fülöp, F. Friedler, A note on the Equivalence of the Process Network Synthesis and Set Covering problems, *Acta Cybernetica* **14**, 2000, 497-502.

[27] Imreh, B., Automaton Theory Approach for Solving PNS Problems, *Acta Cybernetica*, **15**, 2002, 327-338.

[28] Imreh, B., Z. Kovács, A note on separation-networks and automata, *Publicationes Mathematicae*, submitted for publication.

[29] Imreh, Cs., Jól megoldható PNS osztályokról, *Új utak a magyar operációkutatásban, szerk. Komlósi, S., Szántai T., Dialóg Campus Kiadó, Budapest-Pécs*, 1999, 168-181.

[30] Imreh, Cs., A new well-solvable class of PNS problems, *Computing*, **66**, 2001, 289-296.

[31] Karp R. M., Reducibility among Combinatorial Problems in Complexity of Computer Computations, *R. E. Miller and T. W. Thatcher, eds.*, Plenum Press, New York, 1972.

[32] Szpilrajn, E., Sur l'extension de l'ordre partiel, *Fund. Math.* **16** (1930), 386-389.

[33] Volpert, A. I., Differential equations on graphs, *Mat. sb.*, **88**, 1972, 578-588.