ON A WELL-SOLVABLE CLASS OF THE PNS PROBLEM

Z. Blázsik
Research Group on Artificial Intelligence, Hungarian Academy of Sciences
Aradi vértanúk tere 1, H-6720 Szeged, Hungary

Cs. Holló, B. Imreh, Cs. Imreh, Z. Kovács
Department of Informatics, József Attila University,
Árpád tér 2, H-6720 Szeged, Hungary

Abstract

A manufacturing system consists of operating units converting materials of different properties into further materials. In a design problem, we are to find a suitable network of operating units which produces the desired products from the given raw materials. If we consider this network design from structural point of view, then we obtain a combinatorial optimization problem called Process Network Synthesis or (PNS) problem. It is known that the PNS problem is NP-complete. In this work, we present such a subclass of PNS problems which is well-solvable.

AMS Mathematics Subject Classification (1991): 49K35
Key words and phrases: desing problem, PNS problem, Well-solvable problems

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. 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.

The importance of process network synthesis (PNS) arises from the fact that such networks are ubiquitous in the chemical and allied industries. The foundations of PNS and the background of the combinatorial model studied here can be found in [3], [4], [6], [7], and [8]. Therefore, here we shall confine ourselves only to the recall of the necessary definitions.

It has recently been proven (see [1], [5], [9]) that the PNS problem is NP-complete. When a problem is NP-hard or NP-complete, then the studies of some special classes can result in effective procedures for solving the instances of these special classes. A well-known example is the integer linear programming problem which is NP-complete, while such particular cases as the assignment problem or transportation problem can be solved in polynomial time. Another example, the TSP which is NP-complete, but there are some well-solvable subclasses of TSP, a nice overview on them can be found in [2] and [11]. The first well-solvable special classes of PNS problems were studied in [10]. In this work, we present a new subclass of PNS problems which can be solved in polynomial time.

2. Preliminaries

In a combinatorial approach, the structure of a process can be described by the process graph (see [6] and [7]) defined as follows.

Let M be a finite nonempty set, the set of the *materials*. Furthermore, let $\emptyset \neq O \subseteq \wp'(M) \times \wp'(M)$ with $M \cap O = \emptyset$ where $\wp'(M)$ denotes the set of all nonempty subsets of M. The elements of O are called *operating units* and for an operating unit $(\alpha, \beta) \in O$, α and β are called the *input-set* and *output-set* of the operating unit, respectively. The elements of α and β are called the *input* and *output materials* of (α, β), respectively. Furthermore, for every subset S of materials, let us denote by $\Delta(S)$ the set of the operating units having output materials in S. We shall also use the following notations: for any finite set of operating unit o, let

$$mat^{in}(o) = \cup\{\alpha : (\alpha, \beta) \in o\} \quad \text{and} \quad mat^{out}(o) = \cup\{\beta : (\alpha, \beta) \in o\}.$$

The pair (M, O) is defined to be a *process graph* or shortly *P-graph*. The set of vertices of this directed graph is $M \cup O$, and the set of arcs is $A = A_1 \cup A_2$ where $A_1 = \{(X, Y) : Y = (\alpha, \beta) \in O \text{ and } X \in \alpha\}$ 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 the process graph (M, O), then the path determined by these arcs is denoted by $[X_1, X_n]$.

Let the process graphs (m, o) and (M, O) be given. (m, o) is defined to be a *subgraph* of (M, O), if $m \subseteq M$ and $o \subseteq O$.

Now, we can define the structural model of PNS for studying the problem from structural point of view. For this reason, let M^* be an arbitrarily fixed possibly infinite set, the set of the available materials. By *structural model* of PNS, we mean a triplet $\mathbf{M} = (P, R, O)$ where P, R, O are finite sets, $\emptyset \neq P \subseteq M^*$ is the set of the *desired products*, $R \subseteq M^*$ is the set of the *raw materials*, and $\emptyset \neq O \subseteq \wp'(M^*) \times \wp'(M^*)$ is the set of the available operating units. It is assumed that $P \cap R = \emptyset$ and $M^* \cap O = \emptyset$; moreover, α and β are finite sets for every $(\alpha, \beta) = u \in O$.

Then, the process graph (M, O), where $M = \cup\{\alpha \cup \beta : (\alpha, \beta) \in O\}$, represents the interconnections between the operating units of O. Furthermore, every feasible process network, which produces the given set P of products from the given set R of raw materials using operating units from O, corresponds to a subgraph of (M, O). Examining the corresponding subgraphs of (M, O), therefore, we can determine the feasible process networks. If we do not consider further constraints such as material balance, then the subgraphs of (M, O) which can be assigned to the feasible process networks have common combinatorial properties. They are studied in [?], and their description is given by the following definition.

The subgraph (m, o) of (M, O) is called a *solution-structure* of (P, R, O) if the following properties are satisfied:

$(A1)$ $P \subseteq m$,

$(A2)$ $\forall X \in m$, $X \in R \Leftrightarrow$ no (Y, X) arc in the process graph (m, o),

$(A3)$ $\forall Y_0 \in o$, $\exists$ path $[Y_0, Y_n]$ with $Y_n \in P$,

$(A4)$ $\forall X \in m$, $\exists(\alpha, \beta) \in o$ such that $X \in \alpha \bigcup \beta$.

Let us denote the set of solution-structures of (P, R, O) by $S(P, R, O)$.

PNS problem with weigths

Let us consider the PNS problems in which each operating unit has a weight. We are to find a feasible process network with a minimal weight where by weight of a process network we mean the sum of the weights of the operating units belonging to the process network under consideration. Each feasible process network in such a class of PNS problems is determined uniquely from the corresponding solution-structure and vice versa. Thus, the problem can be formalized as follows:

Let a structural model of the PNS problem (P, R, O) be given. Moreover, let w be a positive real-valued function defined on O, the weight function. The basic model is then the following minimization problem:

$$\min\{\sum_{u \in o} w(u) : (m, o) \in S(P, R, O)\}. \tag{1}$$

For the sake of simpicity, in what follows, we call the elements of $S(P, R, O)$ *feasible solutions*, and by a PNS problem we always mean a PNS problem with weights.

3. Hierarchycal PNS problems

A PNS problem is called *hierarchycal* if there exists the partition $M_0 = R, \ldots, M_l = P$ of M and the partition $O_1, \ldots, O_l$, of O such that O_i contains only operating units having input materials from M_{i-1} and output materials from M_i, for all i, $i = 1, \ldots, l$. The hierarchycal PNS problems, which are *thin* in the sense that the size of O_i, $i = 1, \ldots, l$, and the size of M_i, $i = 1, \ldots, l$, are bounded by a fixed constant, are well-solvable. To formulate this statement more precisely, we use the following definition. A PNS problem is called *k-wide hierarchycal* if it is a hierarchycal problem; moreover, $|M_i| \leq k$ and $|O_j| \leq k$ are valid, for all $i = 0, \ldots, l$, $j = 1, \ldots, l$.

Theorem 1. *If a PNS problem is k-wide hierarchycal, then the following procedure either provides an optimal solution of the problem or it gives that the problem has no feasible solution. The time complexity of this algorithm is $C \cdot l$ where C is a constant depending on k.*

Procedure

Subprocedure 1. *(Computing functions F_i and G_i.)*

- *Initialization.* Let N be a number which is greater than $|O| \cdot q$ where q denotes the maximum of the weights of the operating units.
- *Part 0.* Let $G_0(S) = 0$ and $F_0(S) = \emptyset$, for all $S \subseteq M_0$.
- *Part i.* $(i = 1, \ldots, l)$.
  - *Step 1.* If there exists a set $S \subseteq M_i$ for which the functions F_i and G_i have not yet determined, then choose one of them and perform the following steps for it. Otherwise, proceed to the $i+1$-th part if $i < l$, and terminate if $i = l$.
  - *Step 2.* Consider the subset $\Delta(S)$ of O_i and for every set $Q \subseteq \Delta(S)$ examine the validity of $S \subseteq mat^{out}(Q)$. If this relation is false for every Q, then proceed to Step 4. Otherwise, let the sets satisfying the relation above be denoted by $Q_1, \ldots, Q_t$ and proceed to Step 3.
  - *Step 3.* For every Q_j, $j = 1, \ldots, t$, calculate the following value:

$$c_j = G_{i-1}(mat^{in}(Q_j)) + \sum_{u \in Q_j} w(u)$$

    Let us denote a set with a minimal value by Q_j. If there are more sets with the same minimal value, then choose the set having the smallest index. Furthermore, let $F_i(S) = Q_j$, $G_i(S) = c_j$, and proceed to Step 1.
  - *Step 4.* Let $F_i(S) = \emptyset$, $G_i(S) = N$, and proceed to Step 1.

Subprocedure 2. *(For finding an optimal solution)*

- *Initialization.* If $G_l(P) \geq N$, then terminate; the problem has no feasible solution. Otherwise, let $A_0 = P$, $\bar{O}_0 = \emptyset$, and $r = 1$.
- *Iteration (r-th).*
  - *Step 1.* Let $\bar{O}_r = \bar{O}_{r-1} \cup F_{l+1-r}(A_{r-1})$, $A_r = mat^{in}(F_{l+1-r}(A_{r-1}))$. If $r = l$, then proceed to Step 2, otherwise let $r := r + 1$ and proceed to the next iteration.
  - *Step 2.* Terminate; the optimal solution is the P-graph (m, o), where $o = \bar{O}_l$, and $m = mat^{in}(o) \cup mat^{out}(o)$.

Proof. First, we prove that if the algorithm gives a solution, then the produced sets m, o yield a P-graph which is a feasible solution. By the definition of m, it is obvious that for the sets m, o, the P-graph (m, o) exists and satisfies property (A4). Let us observe that if $i < l$, then for each element of A_i, there exists an operating unit in o producing it. This observation follows from the definition of the functions F_j, $j = 1, \ldots, l$. Thus, by $A_0 = P$, we have that (m, o) satisfies property (A1). Since in a hierarchycal PNS problem there is no operating unit producing raw material, we get that in (m, o) there is no edge leading into a raw material. To prove the second part of property (A2), let $X \in m$ be a material with $X \notin R$. Since $X \in m$, thus X is an output or input material of some operating unit from o. In the first case, we get by the definition of the P-graph, that there exists an edge leading into X. In the second case, let $u \in o$ be an operating unit having X as an input material. Since $u \in o$, there exists an index r for which $u \in F_{l+1-r}(A_{r-1})$. This gives that $X \in A_r$. On the other hand, by induction on the number of iterations it is easy to see that $A_i \subseteq M_{l-i}$ for all i, $i = 0, \ldots, l$. This observation results in $r \neq l$. Thus, $X \in A_i$ for some $i < l$ which yields that there exists an edge in (m, o) leading into it. Consequently, property (A2) is also valid for (m, o). To prove property (A3), it is enough to show that for each operating unit from $\bar{O}_i$, $i = 1, \ldots, l$, there exists a path in (m, o) leading from it into a desired product. We prove this statement by induction on i. For the case $i = 1$, we have $A_0 = P$, thus, by the definition of the function F_l, the validity of the statement follows. Now, let $1 \leq i < l$, and let us suppose that the statement is valid for i. We show that it is also valid for $i + 1$. Since $\bar{O}_{i+1} = \bar{O}_i \cup F_{l+1-(i+1)}(A_i)$, thus, by the induction hypothesis, it is enough to prove the statement for the operating units contained in $F_{l+1-(i+1)}(A_i)$. Let $u \in F_{l+1-(i+1)}(A_i)$ be arbitrary. By the definition of the function $F_{l+1-(i+1)}$, we can obtain that u has an output material from the set A_i. (Otherwise, during Step 2 of the construction of the functions, $F_{l+1-(i+1)}(A_i) \subseteq \Delta(A_i)$ is not valid, which is a contradiction.) Let such a material be denoted by Z. By the definition of A_i, it follows that Z is an input material of some operating unit $v \in \bar{O}_i$. Then, by the induction hypothesis, there exists a path $[v, Y]$ in (m, o) where Y is a desired product. Completing the beginning of this path with u and Z, we get a path in (m, o) leading from u into the desired product Y. Thus, we have proved our statement for $i + 1$ which yields that property (A3) is valid for the P-graph (m, o). Consequently, the P-graph determined by the algorithm is a feasible solution.

Now, we prove the correctness of the procedure. To do this, we show first the following statement concerning G_l.

Lemma 1. *For every feasible solution, the weight of the feasible solution is at least $G_l(P)$.*

Proof. Let (m, o) be an arbitrary feasible solution of the problem. Let $o_i = O_i \cap o$, for $i = 1, \dots, l$. Since (m, o) is a feasible solution and the materials of P can be only produced by operating units from O_l, by properties (A1) and (A2), we have that $P \subseteq mat^{out}(o_l)$. The definition of the function G_l and this observation yield the following inequality:

$$G_l(P) \leq G_{l-1}(mat^{in}(o_l)) + \sum_{u \in o_l} w(u).$$

On the other hand, (m, o) is a feasible solution, thus $mat^{in}(o_l) \subseteq m$. The input materials of the operating units from o_l are in the set M_{l-1}, thus, if $l \neq 1$, then they are not contained in R. This yields that for each of them, there exists an operating unit in o having it as an output material. Furthermore, the problem is hierarchical, and hence, the materials from the set M_{l-1} are produced only by operating units from O_{l-1}. These observations yield that $mat^{in}(o_l) \subseteq mat^{out}(o_{l-1})$. This relation and the definition of function G_{l-1} imply the following inequality:

$$G_{l-1}(mat^{in}(o_l)) \leq G_{l-2}(mat^{in}(o_{l-1})) + \sum_{u \in o_{l-1}} w(u).$$

In the same way as above, we obtain that the following inequality is valid, for all i , $i = 1, \ldots, l-1$:

$$G_i(mat^{in}(o_{i+1})) \leq G_{i-1}(mat^{in}(o_i)) + \sum_{u \in o_i} w(u).$$

Summarizing the obtained inequalities, by $G_0(S) = 0$, we get the following inequality:

$$G_l(P) \leq \sum_{i=1}^{l} \sum_{u \in o_i} w(u),$$

which gives the required result.

By Lemma 1, we can prove the correctness of the procedure.

First, we prove that there is no feasible solution of the problem if $G_l(P) \geq N$. Contrary, let us suppose that there is a feasible solution of the problem. Let us denote the weight of this solution by K. By the definition of N, we have that $N > K$. On the other hand, Lemma 1 states that $G_l(P) \leq K$ which results in the contradiction $N > N$.

Now, we show that the feasible solution produced by the algorithm is optimal if $G_l(P) < N$. First, let us observe that the weight of the produced solution is $G_l(P)$. This observation follows immediately from the construction of the algorithm. Thus, by Lemma 1, we obtain that the weight of any feasible solution is at least so large as the weight of the produced solution which means that we get an optimal solution.

Finally, let us examine the time complexity of the procedure. In Subprocedure 1, we perform l parts. During a part, we examine every subsets of $\Delta(S)$, for each subset S of M_i. Since the problem is k-wide hierarchical, M_i has at most 2^k subsets, and since for each such subset S, $\Delta(S) \subseteq O_i$, thus, $\Delta(S)$ can have only 2^k subsets. Consequently, we obtain that the number of operations performed in each iteration is independent on the size of the problem (it depends only on k). In Subprocedure 2, which is based on the functions F_i and G_i, we perform l iterations and the number of operations in each iteration is a constant. This implies that the number of operations performed by the procedure is bounded by $C \cdot l$.

Thus, for every fixed k, the above algorithm solves any k-wide hierarchical problem in linear time. However, we have to note that the constant C in the complexity of the algorithm is exponential in k. This shows that our procedure can be really effective only for small k.

On examining the presented algorithm one can arrive at an interesting observation on the solvability of hierarchical PNS problems.

Corollary 1. *For a hierachical PNS problem, if every material, distinct from the raw materilas, is produced by some operating units, then the problem has a feasible solution.*

Proof. Let us perform the algorithm for the problem. By the above assumption, we obtain that $S \subseteq mat^{out}(\Delta(S))$ for each subset S of materials, which gives that Step 4 is not performed in Subprocedure 1. This yields $G_l(P) < N$, and then the problem has a feasible solution.

References

[1] Blázsik, Z., Imreh, B., A Note on Connection between PNS and Set Covering Problems, Acta Cybernetica, 12 (1996), 309-312.

[2] Burkard, R.E., Deineko, V.G., Van Dal, R., Van der Veen, J.A.A., Woeginger, G.J., Well-Solvable Special Cases of the TSP: A Survey, TU Graz, Bericht Nr. 52 (1995).

[3] Floudas, C.A., Grossmann, I.E., Algorithmic Approaches to Process Synthesis: Logic and Global Optimization, International Symposium on Foundations of Computer Aided Process Design, Snowmass Village, CO, U.S.A., July 10-15, 1994 (in press).

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

[5] Fülöp, J., Imreh, B., Friedler, F., On the reformulation of some classes of PNS problems as set covering problems, Acta Cybernetica 13 (1998), 329-337.

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

[7] Friedler, F., Tarján, K., Huang, Y.W., Fan, L.T., Combinatorial Structure of Process Network Synthesis, Sixth SIAM Conference on Discrete Mathematics, Vancouver, Canada, 1992.

[8] Friedler, F., Tarján, K., Huang, Y.W., Fan, L.T., Graph-Theoretic Approach to Process Synthesis: Polynomial Algorithm for Maximal Structure Generation, Computers chem. Engng. 17(9) (1993), 929-942.

[9] Imreh, B., Fülöp, J., Friedler, F., On the Equivalence of the Set Covering and Process Network Synthesis Problems, in preparation.

[10] Imreh, Cs., Well-solvable PNS classes, XXIII. Hungarian Conference On Operations Research, Pécs, 1997 submitted for publication (In Hungarian).

[11] Lawler, E.L., Lenstra, J.K., Rinnoy kan, A.H.G., Shmoys, D.B., The Traveling Salesman Problem , Wiley, Chichester, 1985, 87-143.