A Note on the Equivalence of the Set Covering and Process Network Synthesis Problems[*]

B. Imreh[†] J. Fülöp[‡] F. Friedler[§]

Abstract

In this paper, combining and completing some earlier results presented in this journal, it is proved that the Process Network Synthesis problem (PNS problem for short) is equivalent to the set covering problem.

Keywords: combinatorial optimization, network design, complexity.

1 Introduction

The foundations of PNS and the background of the combinatorial model studied here can be found in [6], [8], and [9]. Since the present work is based on the results of the papers [3], [7] published in this journal, we recall here only the main definitions briefly.

By *structural model* of PNS we mean a triplet (P, R, O) where P, R, O are finite sets, $\emptyset \neq P$ is the set of *desired products*, R is the set of *raw materials*, $\emptyset \neq O$ is the set of *available operating units*, furthermore $O \subseteq \wp'(M) \times \wp'(M)$ where M is the set of materials involved in the investigation and $\wp'(M)$ denotes the set of all nonempty subsets of M. It is assumed that $P \cap R = \emptyset$ and $M \cap O = \emptyset$. An operating unit, $u = (\alpha, \beta) \in O$, can be interpreted such that α and β contain the *input* and *output* materials of u, respectively. Pair (M, O) determines a directed graph called *process graph.* The set of vertices of this 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 process graph (M, O), then

[*]This work has been supported by the Ministry of Education of Hungary, Grant FKFP 0008/1999. The research of the second author was supported in part by Hungarian National Research Foundation, Grant OTKA T016413.

[†]Department of Informatics, József Attila University, Árpád tér 2, H-6720 Szeged, Hungary

[‡]Laboratory of Operations Research and Decision Systems, Computer and Automation Institute, Hungarian Academy of Sciences, H-1518 Budapest, P.O.Box 63, Hungary

[§]Department of Computer Science, University of Veszprém, Egyetem u. 10, H-8200 Veszprém, Hungary

the path determined by these arcs is denoted by $[X_1, X_n]$. Obviously, for any suitable pair (m, o), one can consider the corresponding process graph.

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

For structural model (P, R, O), process graph (M, O) with $M \supseteq \cup\{\alpha \cup \beta : (\alpha, \beta) \in O\}$ presents the interconnections among the operating units of O. Furthermore, every feasible process network, producing 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). Therefore, examining the corresponding subgraphs of (M, O), we can determine the feasible process networks. If there is no further constraints such as material balance, then the subgraphs of (M, O) which can be assigned to the feasible process networks have the following common combinatorial properties (*cf.* [8]).

Subgraph (m, o) of (M, O) is called a *solution-structure* of (P, R, O) if:

$(S1)$ $P \subseteq m$,

$(S2)$ $\forall X \in m$, $X \in R \Leftrightarrow X$ is a source in (m, o),

$(S3)$ $\forall Y_0 \in o$, $\exists$ path $[Y_0, Y_n]$ with $Y_n \in P$,

$(S4)$ $\forall X \in m$, $\exists(\alpha, \beta) \in o$ such that $X \in \alpha \bigcup \beta$.

Let $S(P, R, O)$ denote the set of solution-structures of (P, R, O). It can be easily seen that any solution-structure (m, o) is uniquely determined by set o. By this observation, $S(P, R, O)$ can be considered as a family of subsets of O.

PNS problem with weigths

Let a PNS problems be given. Let us suppose that its every operating unit has a positive weight. Then, the following optimization problem can be studied:

We are to find a feasible process network with the 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 PNS problem is determined uniquely from the corresponding solution-structure and vice versa. Thus, the problem can be formalized as follows:

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

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

In what follows, the elements of $S(P, R, O)$ are called *feasible solutions* and by PNS problem we mean a PNS problem with weights.

2 Relationship between the PNS and the set covering problems

We recall (*cf.* [3]) that any set covering problem can be transformed into an equivalent PNS problem. On the other hand, it is proved in [7] that every cycle free PNS problem can be transformed into a suitable set covering problem. Now, using a similar argument as in [7], we extend this statement to an arbitrary PNS problem which results in the equivalence of the two problems considered.

This extension is proved in more steps.

First, let us observe that $S(P,R,O)$ is closed under the finite union. This results in the existence of a greatest feasible solution, $\cup\{(m,o) : (m,o) \in S(P,R,O)\}$, which is called *maximal structure* of the problem provided that $S(P,R,O) \neq \emptyset$. The existence of the maximal structure makes possible a reduction of PNS problem under consideration in the following way. Let a PNS problem be given by (P,R,O) and let us denote (M,O) the process graph belonging to it. Furthermore, let us suppose that $S(P,R,O) \neq \emptyset$, and let denote $(\bar{M},\bar{O})$ the corresponding maximal structure. Then, we can construct a new model given by $(P,R\cap\bar{M},\bar{O})$. Since $(\bar{M},\bar{O})$ contains every feasible solution from $S(P,R,O)$, $S(P,R,O) = S(P,R\cap\bar{M},\bar{O})$. This new model is called the *reduced model* of problem considered. By the basis of the common set of feasible solutions, we obtain that problem

$$(2) \qquad \min\{\sum_{u\in o} w(u) : (m,o) \in S(P,R\cap\bar{M},\bar{O})\}.$$

is equivalent to problem (1) provided that $S(P,R,O) \neq \emptyset$.

Now, if $S(P,R,O) = \emptyset$, then the set covering problem consisting of set P and a nonempty proper subset of P with an arbitrary weight is equivalent to the PNS problem under consideration, since both problems have no feasible solution. On the other hand, $S(P,R,O) = \emptyset$ can be decided in polynomial time by using the algorithm presented in [9] for generating the maximal structure of (P,R,O).

If $S(P,R,O) \neq \emptyset$, then instead of (1) we can consider (2) where the process graph of the problem is its maximal structure $(\bar{M},\bar{O})$. In this case, a conjuctive normal form (CNF) exists for describing the feasible solutions. Description of the feasible solutions by (CNF) was originally initiated in [4] and this description is also used in [7]. Here, taking the maximal structure into account, a simplest and more precise description is given. The basis of this approach is the observation that a faesible solution is determined by o uniquely. This observation makes possible to the reformulation of properties $(S1)$ through $(S4)$.

To do this, let $\bar{O} = \{(\alpha_1,\beta_1),\dots,(\alpha_l,\beta_l)\}$, and $J = \{1,\dots,l\}$. Then, for any subgraph (m,o) of $(\bar{M},\bar{O})$, an l-vector of logical values y_i, $i \in J$, can be associated with such that y_i is true if and only if $(\alpha_i,\beta_i) \in o$. It is easy to see that this is a one-to-one mapping between the subgraphs of $(\bar{M},\bar{O})$ fulfilling $(S4)$ and the l-vectors of logical values. For logical l-vector $\mathbf{y}$, subgraph (m,o) associated with $\mathbf{y}$ is determined by $m = \bigcup_{i\in T(\mathbf{y})} \alpha_i \cup \beta_i$ and $o = \{(\alpha_i,\beta_i) : i \in T(\mathbf{y})\}$ where

$T(\mathbf{y}) = \{i : i \in J \;\&\; y_i \text{ is true}\}$. Obviously, for an arbitrary logical l-vector, $\mathbf{y}$, the subgraph associated with $\mathbf{y}$ is not necessarily a feasible solution. Procedure below provides such a CNF, Φ, that a logical l-vector, $\mathbf{y}$, satisfies Φ if and only if the subgraph associated with $\mathbf{y}$ is a feasible solution.

Algorithm for CNF Generation

- *Step 1.* Set $\Phi_0 = \bigwedge_{X \in P} \bigvee_{\substack{i \in J \\ X \in \beta_i}} y_i$.
- *Step 2.* Let $\Phi_1 = \bigwedge_{\substack{i \in J \\ X \in \alpha_i \setminus R}} (\neg y_i \vee \bigvee_{\substack{h \in J \\ X \in \beta_h}} y_h)$.
- *Step 3.* Set $\Phi_2 = \bigwedge_{\substack{i \in J \\ P \cap \beta_i = \emptyset}} (\neg y_i \vee \bigvee_{\substack{h \in J \\ \beta_i \cap \alpha_h \neq \emptyset}} y_h)$.
- *Step 4.* Let $\Phi = \Phi_0 \wedge \Phi_1 \wedge \Phi_2$.

Now, we show that an l-vector $\bar{\mathbf{y}}$ of logical values satisfies Φ if and only if process graph associated with $\bar{\mathbf{y}}$ is a feasible solution. To do this, let (m, o) be an arbitrary feasible solution in $(\bar{M}, \bar{O})$, and let us denote by $\bar{\mathbf{y}}$ the l-vector of logical values associated with (m, o). Then, $(S1)$ implies $\Phi_0(\bar{\mathbf{y}}) =\uparrow$ and $(S2)$ implies $\Phi_1(\bar{\mathbf{y}}) =\uparrow$. Regarding Φ_2, let $i \in J$ and $P \cap \beta_i = \emptyset$. Then, $\neg y_i \bigvee_{\substack{h \in J \\ \beta_i \cap \alpha_h \neq \emptyset}} y_h$ is a member of Φ_2. If $\bar{y}_i =\downarrow$, then the considered disjunction is true. If $\bar{y}_i =\uparrow$, then $u_i \in o$, but in this case, $(S3)$ implies that there is an operating unit $u_h \in o$ such that $\beta_i \cap \alpha_h \neq \emptyset$. This yields that $\bar{y}_h =\uparrow$ for some $h \in J$, and thus, the considered member is also true. Idea presented above is valid for every member of Φ_2, and hence, $\Phi_2(\bar{\mathbf{y}}) =\uparrow$. Consequently, $\bar{\mathbf{y}}$ satisfies Φ.

Conversely, let us suppose that the logical vector $\bar{\mathbf{y}}$ satisfies Φ. Let (m, o) denote the subgaph of $(\bar{M}, \bar{O})$ associated with $\bar{\mathbf{y}}$. We prove that (m, o) is a feasible solution, *i.e.*, properties $(S1)$ through $(S4)$ hold for (m, o).

Property $(S4)$ follows directly from the definition of (m, o).

Since $\Phi_0(\bar{\mathbf{y}}) =\uparrow$, there exists an operating unit in o which produces X directly, for every $X \in P$. Consequently, $(S1)$ is valid as well.

To prove $(S2)$, let $X \in m$ and $X \in R$. Since $o \subseteq \bar{O}$ and there is no operating unit in $\bar{O}$ producing raw material, there is no (Y, X) arc in (m, o). Now, let us suppose that $X \in m$ and there is no (Y, X) arc in (m, o). Then, we show that $X \in R$. Contrary, let us assume that $X \notin R$. Since $X \in m$, there exists an operating unit, $u_i = (\alpha_i, \beta_i) \in o$ such that $X \in \alpha_i$ which implies $\bar{y}_i =\uparrow$. On the other hand, since $(\bar{M}, \bar{O})$ is a feasible solution, there are operation units $u_h \in \bar{O}$ such that $X \in \beta_h$, and therefore, Φ_1 contains a member of the form $\neg y_i \bigvee_{\substack{h \in J \\ X \in \beta_h}} y_h$. Since $\Phi_1(\bar{\mathbf{y}}) =\uparrow$, this member is also true on $\bar{\mathbf{y}}$, and hence, there is an h_0 such that

$u_{h_0} \in o$ and $X \in \beta_{h_0}$. This yields that there exists (Y, X) arc in (m, o) which is a contradiction. Therefore, $X \in R$, and thus, $(S2)$ is valid.

For proving $(S3)$, let us suppose that $u_i \in o$. If $\beta_i \cap P \neq \emptyset$, then we are ready, there is a path from u_i into P. In the opposite case, using again that $(\bar{M}, \bar{O})$ is a feasible solution, there is a path from u_i into P in $(\bar{M}, \bar{O})$, *i.e.*, there is an operating unit $u_h \in \bar{O}$ such that $\beta_i \cap \alpha_h \neq \emptyset$. Then, Φ_2 contains a member of form $\neg y_i \bigvee_{\substack{h \in J \\ \beta_i \cap \alpha_h \neq \emptyset}} y_h$, and in a similar way as above, we obtain that $u_h \in o$ for some h. Repeating this procedure, in a finite number of steps, we obtain a path from u_i into P in (m, o), and therefore, $(S3)$ is valid.

By the strong relationship between the PNS problem and the corresponding CNF given above, the following problem is equivalent to problem (2).

$$(3) \qquad \min\{\sum_{j \in T(\mathrm{y})} w_j : \mathrm{y} \text{ fullfils } \Phi\}$$

where $w_j = w(u_j)$, $j = 1, \ldots, l$.

For every $i \in J$, introducing two 0-1 variables, z_i^+ and z_i^-, such that $z_i^+ = 1$ if and only if y_i is true, and $z_i^- = 1 - z_i^+$, problem (3) can be transcribed into an equivalent binary programming problem by introducing some appropriate new constraints. It is easy to see that this binary problem is a set covering/partitioning problem. On the other hand, using the well-known trick of converting set partitioning constraints into set covering ones (*cf. e.g.* [10]), we obtain an equivalent set covering problem. The corresponding binary and set covering problems can be found in [7].

Summarizing, we obtained that the solution of a PNS problem can be traced back to the solution of a suitable set covering problem. Combining this statement by the observation that any set covering problem can be given as a special PNS problem, we obtain the following statement.

Proposition 1. *The PNS problem is equivalent to the set covering problem.*

Equivalence proved above enables the sophisticated techniques developed for solving set covering problems (see, *e.g.*, [2, 5, 10] and the references therein) also to be applied for solving PNS problems. Regarding this solution technique, it is an open problem whether a more economical transformation of the PNS problem into a set covering problem exists.

It is worth noting that Proposition 1, by the well-known fact that the set covering problem is NP-complete (see [1] and [11]), implies immediately the following observation.

Corollary 1. *The PNS problem is NP-complete.*

References

[1] A. V. Aho, J. E. Hopcroft, J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading Mass, 1974.

[2] E. Balas, M. C. Carrera, A Dynamic Subgradient- Based Branch-and-Bound Procedure for Set Covering, *Operations Research* 44 (1996), 875-890.

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

[4] M. H. Brendel, F. Friedler and L.T. Fan, Combinatorial Foundation for Logical Formulation in Total Flowsheet Synthesis, *Computers chem. Engng.*, submitted for publication.

[5] M. L. Fisher, P. Keida, Optimal Solution of Set Covering/Partitioning Problems Using Dual Heuristics, *Management Science* **36** (1990), 674-688.

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

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

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

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

[10] R. S. Garfinkel, G. L. Nemhauser, *Integer Programming*, Wiley, New York, 1972.

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

Received October, 1999