NOVI SAD J. MATH.
VOL. 30, NO. 3, 2000, 11-19

ON BOTTLENECK AND k-SUM VERSIONS OF THE PROCESS NETWORK SYNTHESIS 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

In this paper, we deal with a special case of the Minsum problem, the Process Network Synthesis (PNS) problem. We show that the k-sum version of the PNS problem is well-solvable, and thus, the PNS problem is such a particular case of the Minsum problem which is NP-complete while its k-sum version is well-solvable for a fixed k.

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

1. Minsum, Bottleneck and k-sum Optimization Problems

Let E be a finite set, F be a family of subsets of E, and $f : F \to \Re$ be a function assigning a real number to each $S \in F$. Then, a general combinatorial optimization problem is to find an $S^* \in F$ with $f(S^*) \leq f(S)$,

for all $S \in F$. In most combinatorial optimization problems, there is a weight function $c : E \to \Re$, and the object function $f(S)$ is defined in terms of these weights in different ways:

When $f(S) = \max\{c_e : e \in S\}$, then we obtain the *Minmax* or *Bottleneck Optimization problem*, (BOP) in short,

If $f(S) = \sum_{e \in S} c_e$, then we get the *Minsum problem*, or (MSP) in short,

where c_e always denotes the value of $c(e)$ for the sake of simplicity. Since several NP-complete problems are known to be particular cases of (BOP) and (MSP), both problems are NP-hard. Efficient solution procedures are known when F is specially structured, for example in the case of the Assignment problem.

For an $S \in F$, let us suppose that the elements of S are written in a non-increasing order with respect to their weights. More precisely, $S = \left\{s_1, \ldots, s_{|S|}\right\}$ and $c_{s_i} \geq c_{s_{i+1}}$, for $i = 1, \ldots, |S| - 1$. Now, for any given integer $k \geq 1$ let $f_k(S) = \sum_{i=1}^{p} c_{s_i}$, where $p = \min\{|S|, k\}$. Then, the *k-sum Optimization problem* (SUM(k) in short) is defined in the following way:

$$\min\{f_k(S) : S \in F\}.$$

When $k = 1$, the SUM(k) problem is reduced to the (BOP) and when $k \geq \max\{|S| : S \in F\}$, it is reduced to the (MSP). Thus, SUM(k) simultaneously generalizes both (BOP) and (MSP); and if either (BOP) or (MSP) is NP-complete, then so is SUM(k).

In general, SUM(k) problem was studied by Gupta and Punnen [7] and later by Punnen and Aneja [10], and it was shown that SUM(k) can be solved by solving $O(m)$ Minsum problems ($m = |E|$). Thus, if we have a polynomial-time algorithm to solve the Minsum problem, then we have a polynomial-time algorithm to solve the SUM(k) problem for any k, ($1 \leq k \leq m$). In the case of the PNS problem, the Minsum problem is NP-complete (see [9]), and thus, the k-sum version of the PNS problem under an arbitrary k is NP-complete, too.

2. Basic definition of PNS problem

The introduction of the PNS problem and its combinatorial model can be found in the works [3], [4], [5], and [8]. Here, we recall only the necessary

definitions.

Let $M \neq \emptyset$ be a finite 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 pair (M, O) is called a *process graph* or P-*graph* in short. The set of its vertices is $M \cup O$, and the set of its arcs is $A = A_1 \cup A_2$ where $A_1 = \{(X, Y) : Y = (\alpha, \beta) \in O \ \& \ X \in \alpha\}$ and $A_2 = \{(Y, X) : Y = (\alpha, \beta) \in O \ \& \ X \in \beta\}$. If $X_1, X_2, ..., X_n$ are vertices such that $(X_1, X_2), (X_2, X_3), \ldots, (X_{n-1}, X_n)$ are arcs of (M, O), then the path consisting of these arcs is denoted by $[X_1, X_n]$.

Let $o \subseteq O$ be arbitrary, and let us define the functions below on the set o

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

and

$$mat(o) = mat^{in}(o) \bigcup mat^{out}(o).$$

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

Now, we are ready to define the structural model of PNS. For this purpose, let M be an arbitrarily fixed finite set, the set of the available materials. By *structural model* of PNS, we mean a system (P, R, O), where $\emptyset \neq P \subseteq M$ is the set of the *desired products*, $R \subseteq M$ is the set of the *raw materials*, and $O \subseteq \wp'(M) \times \wp'(M)$ is the set of the available operating units. It is supposed that $P \cap R = \emptyset$ and $M \cap O = \emptyset$. In this case, the process graph (M, O) represents the interconnections between the operating units of O, where $M = \cup\{\alpha \cup \beta : (\alpha, \beta) \in O\}$.

Moreover, 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). Therefore, by investigating the corresponding subgraphs of (M, O), one can determine the feasible process networks. If we do not take into account further constraints such as material balance, then the subgraphs of (M, O) which can be assigned to the feasible process networks can be described by some combinatorial properties. These properties are established in [4] and they are presented by the following definition.

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

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

$(A2)$ $\forall X \in m$, $X \in R \Leftrightarrow$ 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 $\mathbf{M} = (P, R, O)$ by $S(P, R, O)$ or $S(\mathbf{M})$.

PNS problem with weights

Let us consider such PNS problems in which every operating unit has a positive weight. We are to find a feasible process network with 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. Therefore, the problem can be formalized in the following way:

Let $\mathbf{M} = (P, R, O)$ be given; moreover, let w be a positive real-valued function defined on O, the weight function. Then, the basic model is as follows:

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

In what follows, for the sake of simplicity, the elements of $S(\mathbf{M})$ are called *feasible solutions* and by PNS problem we always mean a PNS problem with weights. It is known (see [2], and [9]) that problem (1) is NP-complete. Furthermore, it is easy to see that the set o determines the P-graph (m, o) uniquely, for every feasible solution (m, o). Consequently, (1) is a particular case of the Minsum problem.

It is a basic observation that if (m, o) and (m', o') are feasible solutions of $\mathbf{M}$, then $(m, o) \cup (m', o')$ is also a feasible solution of $\mathbf{M}$. This implies that $S(\mathbf{M})$ has a greatest element called *maximal structure* provided that $S(\mathbf{M}) \neq \emptyset$. Indeed, the maximal structure is the union of all the feasible solution

in $\mathbf{M}$. Obviously, the P-graph of an arbitrary PNS problem can contain unnecessary operating units and materials. On the basis of the maximal structure, we can disregard these unnecessary operating units and materials as follows. Let $(\bar{M}, \bar{O})$ denote the P-graph of the maximal structure. Then, the P-graph of the structural model $\bar{\mathbf{M}} = (P, R \cap \bar{M}, \bar{O})$ is $(\bar{O}, \bar{M})$, and since each feasible solution of $\mathbf{M}$ is a subgraph of $(\bar{M}, \bar{O})$, it is a feasible solution of $\bar{\mathbf{M}}$, and conversely. Consequently, $S(\mathbf{M}) = S(\bar{\mathbf{M}})$. On the other hand, $\bar{\mathbf{M}}$ does not contain any unnecessary operating unit and material. The structural model $\bar{\mathbf{M}}$ is called *reduced structural model* of PNS.

To determine the reduced structural model for a PNS problem, an effective procedure is presented in [6], [5]; it can decide if $S(\mathbf{M})$ is empty; if $S(\mathbf{M})$ is not empty, then the algorithm provides the corresponding maximal structure in polynomial time. It is a simple observation [1], that if we take a subset of the operating units, then we have the input and output materials of these operating units. In other words, if we have an operating unit type vertex set o of the bipartite process graph (M, O), then the set o determines a subgraph (m, o) of (M, O) where $m = mat(o)$, and the arcs of this subgraph are the same ones as the arcs between o and m in (M, O). By our Algorithm for Maximal Structure Generation (AMSG), we can decide whether there is a feasible solution (m', o') of $\mathbf{M}$ in (m, o) or not.

3. Bottleneck PNS-problem

Let a reduced structural model of PNS problem $\mathbf{M} = (P, R, O)$ be given. We are to find such a feasible solution in which the weightest operating unit has the least weight. Formally, we are to solve

$$\min\{\max\{w(u) : u \in o\} : (m, o) \in S(\mathbf{M})\}. \tag{2}$$

To solve (2), let $O = \{u_1, \ldots, u_n\}$. Without loss of generality, it can be supposed that $w(u_1) \leq w(u_2) \leq \ldots \leq w(u_n)$. For every positive integer $i(\leq n)$, let $O_i = \{u_1, \ldots, u_i\}$ and $M_i = mat(O_i)$. Furthermore, let $\mathbf{M}_i = (P, R, O_i)$. Then, the following statement is valid.

Lemma 1. *If for some integer* $1 \leq i \leq n$, $S(\mathbf{M}_i)$ *has the maximal structure and* $S(\mathbf{M}_{i-1})$ *has no the maximal structure, then* u_i *is included in every feasible solution in* $S(\mathbf{M}_i)$.

By the above statement, we can show that the optimal value of (2) is $w(u_i)$. Indeed, let us consider an arbitrary feasible solution $(m', o') \in S(\mathbf{M})$. If there is a $j > i$ with $u_j \in o'$, then $w(u_i) \leq \max\{w(u) : u \in o'\}$. Now, let us suppose that $u_j \in o'$ implies $j \leq i$. If $u_i \in o'$, then $w(u_i) = \max\{w(u) : u \in o'\}$. Finally, $u_i \notin o'$ is impossible, since (m', o') is a feasible solution of $\mathbf{M}_i$, and thus, by Lemma 1, $u_i \in o'$. Consequently,

$$w(u_i) = \min\{\max\{w(u) : u \in o\} : (m.o) \in S(\mathbf{M})\}.$$

Now, we can solve (2) by the following procedure.

Procedure 1.

- *Initialization.* Let $i = 1$ and $O_i = \{u_1\}$.
- *Iteration (i-th).* Let $M_i = mat(O_i)$. Let us perform the Algorithm for Maximal Structure Generation for the structural model $\mathbf{M}_i = (P, R, O_i)$. If there exists the maximal structure, then terminate; the maximal structure is an optimal solution and the optimal value is $w(u_i)$. In the opposite case, let $O_{i+1} = \{u_1, \ldots, u_{i+1}\}$, $i := i + 1$, and proceed to the next iteration.

It is easy to see that the time complexity of this procedure is $n \cdot q$ where q denotes the time complexity of the Algorithm for Maximal Structure Generation.

Remark. We note that this time complexity can be improved by changing the performance of the procedure. It is easy to see that one can start with $i = [n/2]$, and if there is the maximal structure, then, as the next step, one can choose the middle point of $[1, i]$, and in the opposite case, the middle point of $[i, n]$. This performance provides a better time complexity, which is $q \cdot \log(n)$, where q denotes the same constant as above.

4. k-sum version of PNS problem

Let (M, O) be the maximal structure of a reduced structural model of PNS problem $\mathbf{M} = (P, R, O)$. Furthermore, let k be a fixed positive integer. We are to find such a feasible solution for which the sum of weights of the k weightest operating units is minimal.

To formalize the problem considered, let us denote by $u_{i_1}, \ldots, u_{i_k}$, the k weightest operating units of (m, o). The *k-sum* PNS *problem* is then

$$\min\{\sum_{t=1}^{k} w(u_{i_t}) : (m, o) \in S(\mathbf{M})\}, \tag{3}$$

where by if a feasible solution has less operating units than k, we equip the weightest operating units with the copies of a fictious operating unit having 0 weight.

Since o determines the P-graph (m, o) uniquely, for every feasible solution, (3) is a particular case of the SUM(k) problem.

For solving (3), Let $O = \{u_1, \ldots, u_n\}$. Without loss of generality, we can assume again that $w(u_1) \leq w(u_2) \leq \ldots \leq w(u_n)$. Let us fix now such a linear ordering, denoted by $\preceq$, on the subsets of at most k elements of O for which

$$\{u_{i_1}, \ldots, u_{i_r}\} \preceq \{u_{j_1}, \ldots, u_{j_s}\} \text{ if and only if } \sum_{t=1}^{r} w(u_{i_t}) \leq \sum_{t=1}^{s} w(u_{i_t}).$$

Such an ordering there exists; moreover, it can be determined by some rules.

For every subset $\{u_{i_1}, \ldots, u_{i_r}\} \subseteq O$, let

$$O_{\{u_{i_1}, \ldots, u_{i_r}\}} = \{u_{i_1}, \ldots, u_{i_r}\} \cup \{u_t : u_t \in O \;\&\; w(u_t) \leq w(u_{i_1})\},$$

where it is supposed that u_{i_1} has the smallest index in $\{u_{i_1}, \ldots, u_{i_r}\}$. Furthermore, let $\mathbf{M}_{\{u_{i_1}, \ldots, u_{i_r}\}} = (P, R, O_{\{u_{i_1}, \ldots, u_{i_r}\}})$. Then, the following assertion is valid.

Lemma 2. *If for some* $\{u_{i_1}, \ldots, u_{i_r}\} \subseteq O$, $S(\mathbf{M}_{\{u_{i_1}, \ldots, u_{i_r}\}})$ *has the maximal structure and for every subset* $\{u_{j_1}, \ldots, u_{j_s}\} \subseteq O$ *with* $\{u_{j_1}, \ldots, u_{j_s}\} \preceq \{u_{i_1}, \ldots u_{i_r}\}$, $S(\mathbf{M}_{\{u_{j_1}, \ldots, u_{j_s}\}})$ *has no the maximal structure, then the maximal structure of* $S(\mathbf{M}_{\{u_{i_1}, \ldots, u_{i_r}\}})$ *is an optimal solution of (3), and the optimal value is* $\sum_{t=1}^{r} w(u_{i_t})$.

On the basis of Lemma 2, one can determine an optimal solution of (3) by the following procedure.

Procedure 2.

- *Step 1.* Establish the corresponding linear ordering.

- *Step 2.* Let $i = 1$.
- *Step 3.* Consider the i-th subset of O regarding the fixed ordering. Let $\{u_{i_1}, \ldots, u_{i_r}\}$ be this subset. Perform the Algorithm for Maximal Structure Generation for $\mathbf{M}_{\{u_{i_1}, \ldots, u_{i_r}\}}$. If there exists the maximal structure, then terminate; the maximal structure is an optimal solution. In the opposite case, let $i := i + 1$ and repeat Step 3.

The time complexity of this procedure is $\sum_{t=1}^{k} \binom{n}{t} \cdot q$, where q denotes the time complexity of the Algorithm for Maximal Structure Generation.

It is worth noting that the technique presented in this section is suitable to solve a generalized version of (3). Namely, if the object function is not the sum of the weights of the k weightest operating units but it only depends on them, *i.e.*, it has a form $z(w(u_{i_1}), \ldots, w(u_{i_k}))$, where fictious operating units are allowed, then we obtain the following problem:

$$\min\{z(w(u_{i_1}), \ldots, w(u_{i_k})) : (m, o) \in S(\mathbf{M})\}, \tag{4}$$

In this case, the linear ordering has to satisfy the following condition:

$$\{u_{i_1}, \ldots, u_{i_r}\} \preceq \{u_{j_1}, \ldots, u_{j_s}\} \text{ iff } z(w(u_{i_1}), \ldots, w(u_{i_k})) \leq z(w(u_{j_1}), \ldots, w(u_{j_k})).$$

The bottleneck PNS-problem is well-solvable, but the PNS problem is NP-complete (*cf.* [2], [9]). Our method for solving the k-sum version of PNS-problem is polynomial for "small" fixed k, similarly to the example of minimization of the sum of k tardinesses in scheduling, was examined by Woeginger [11].

References

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

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

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

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

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

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

[7] Gupta, S.K., Punnen, A.P., k-sum optimization problems, Oper. Res. Letters 9 (1990), 121-126.

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

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

[10] Punnen A.P., Aneja, Y.P., On k-sum optimization, Operations Research Letters 18 (1996), 233-236.

[11] Woeginger, G., On minimizing the sum of k tardinesses, Inform. Process. Letters 38 (1991), 253-256.