Acta Cybernetica 15 (2002) 327–338.

Automaton Theory Approach for Solving Modified PNS Problems*

B. Imreh†

To Professor Masami Ito on his 60th birthday

Abstract

In this paper a modified version of the Process Network Synthesis (PNS) problem is studied. By using an automaton theoretical approach, a procedure for finding an optimal solution of this modified PNS problem is presented.

Introduction

The Process Network Synthesis (PNS for short) problem can be considered as a particular process design optimization. For this design, a set of the available operating units is given and each operating unit has a positive weight. Moreover, two distinguished sets of materials, the sets of raw materials and required products are also given. We are to find a minimum-weight process, consisting of the available operating units, which produces the required products from the raw materials. The corresponding processes from structural point of view can be identified by particular bipartite graphs satisfying some conditions. Such conditions are established in [4] and [5]. The bipartite graphs satisfying these conditions are called *solution-structures* and they can be considered as generalized feasible processes. This generalization means that we consider the processes in dynamic sense when we do not require the executability of processes. Therefore, a solution-structure may represent a non-executable process where by the *executability* of a process we mean that there exists such a scheduling of its operating units that the process can be performed in accordance with this scheduling. Here, by introducing a new condition for the bipartite graphs, we modify the original problem, concerning the generalized feasible processes, to such one whose feasible solutions represent exactly the executable feasible processes. For solving this modified problem, we extend the idea of [8]. Namely, for every instance of the modified problem, we define such an

*This work has been supported by the Hungarian National Foundation for Scientific Research, Grant T037258.

†Department of Informatics, University of Szeged, Árpád tér 2, H-6720 Szeged, Hungary

automaton that an optimal solution can be found by performing a shortest path method in the weighted transition graph of this automaton.

The paper is organized as follows. In Section 1, we recall the PNS problem and introduce its modified version, moreover, we recall some necessary notions and notation on automata. Then, in Section 2, the automaton theoretical approach and a procedure for finding an optimal solution are presented.

1 Preliminaries

Since the description of the original PNS problem can be found in more works (see *e.g.* [4], [5], [6], and [7]), we recall only the necessary definitions here. In the combinatorial approach, the structure of a process can be described by the process graph (*cf.* [5]) defined as follows.

Let M be a finite nonempty set, the set of the *materials*, and 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 any operating unit $u = (C, D) \in O$, C and D are called the *set of the input and output materials* of u, respectively. The pair (M, O) is called a *process graph*. The set of vertices of (M, O) is $M \cup O$, and the set of arcs is $E = E_1 \cup E_2$, where $E_1 = \{(x, u) : u = (C, D) \in O \; \& \; x \in C\}$ and $E_2 = \{(u, x) : u = (C, D) \in O \; \& \; x \in D\}$. If there are vertices $x_1, x_2, ..., x_n$, such that $(x_1, x_2), (x_2, x_3), \dots, (x_{n-1}, x_n)$ are arcs of (M, O), then the path belonging to these arcs is denoted by $path[x_1, x_n]$. Let the process graphs $(\bar{M}, \bar{O})$ and (M, O) be given: $(\bar{M}, \bar{O})$ is called a *subgraph* of (M, O), if $\bar{M} \subseteq M$ and $\bar{O} \subseteq O$.

For any $\bar{O} \subseteq O$, let us define the following functions on $\bar{O}$:

$$mat^{in}(\bar{O}) = \bigcup_{(C,D)\in\bar{O}} C, \qquad mat^{out}(\bar{O}) = \bigcup_{(C,D)\in\bar{O}} D,$$

and

$$mat(\bar{O}) = mat^{in}(\bar{O}) \cup mat^{out}(\bar{O}).$$

Now, we can define the instances of the process design problem as follows. By an *instance* of the process design problem we mean a quartet $\mathbf{M} = (M, O, P, R)$, where M, O, P, R are finite sets; M is the set of the *available materials*, $\emptyset \neq P \subseteq M$ 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 supposed that $P \cap R = \emptyset$ and $M \cap O = \emptyset$. We are to design a process from structural point of view which produces the given set P of the required products from the given set R of the raw materials by using some available operating units.

Let us observe that the process graph (M, O) describes the interconnections among the operating units of O. Furthermore, every generalized feasible process corresponds to a subgraph of (M, O). Consequently, we can determine the generalized feasible processes by examining the corresponding subgraphs of (M, O). If we do not consider further constraints such as material balance, then the subgraphs of (M, O) which can be assigned to the generalized feasible processes have common combinatorial properties. Such properties are established in [4] and [5]. A subgraph $(\bar{M}, \bar{O})$ of (M, O) is called a *solution-structure* of $(M, O, P, R,)$ if the following conditions are satisfied:

$(A1)$ $P \subseteq \bar{M}$,

$(A2)$ $\forall x \in \bar{M}$, $x \in R \Leftrightarrow$ no (u, x) arc in the process graph $(\bar{M}, \bar{O})$,

$(A3)$ $\forall u \in \bar{O}$, $\exists$ $path[u, x]$ with $x \in P$,

$(A4)$ $\forall x \in \bar{M}$, $\exists (C, D) \in \bar{O}$ such that $x \in C \cup D$.

The set of the solution-structures of $\mathbf{M}$ is denoted by $S(M, O, P, R,)$ or $S(\mathbf{M})$. We shall use the following observation which can be easily proved.

Remark 1. *If $(\bar{M}, \bar{O})$ is a solution-structure of $\mathbf{M}$, then $\bar{M} = mat(\bar{O})$, and hence, $\bar{O}$ determines the solution-structure $(\bar{M}, \bar{O})$ uniquely.*

Let us now consider an instance of the process design problem in which every operating unit has a positive real weight. We are to find a solution-structure with minimal weight where by the *weight of a process graph* we mean the sum of the weights of the operating units belonging to the process graph under consideration. Now, an optimization problem, called PNS *problem*, can be formalized in the following way:

Let an instance $\mathbf{M} = (M, O, P, R)$ of the process design problem be given. Moreover, let w be a positive real-valued function defined on O, the *weight function*. The optimization problem is then

$$(1) \qquad \min\{\sum_{u \in \bar{O}} w(u) : (\bar{M}, \bar{O}) \in S(M, O, P, R)\}.$$

It is worth noting that the PNS problem is NP-hard (*cf.* [1]).

As we mentioned some of the feasible solutions of (1), which are solution-structures, may represent non-executable processes. To illustrate this fact, let us consider the following simple example.

Example 1. Let $M = \{a_1, \ldots, a_6\}$, $R = \{a_1, a_2\}$, $P = \{a_6\}$ and $O = \{u_1, u_2, u_3\}$, where the definition of the operating units are given by the table below.

	input materials	output materials
u_1	a_1, a_2	a_3
u_2	a_3, a_4	a_5
u_3	a_5	a_4, a_6

Now, $S(M, O, P, R)$ is a singleton set and the only one solution-structure represent such a process which can not be executed. The corresponding process graph is depicted in Figure 1.

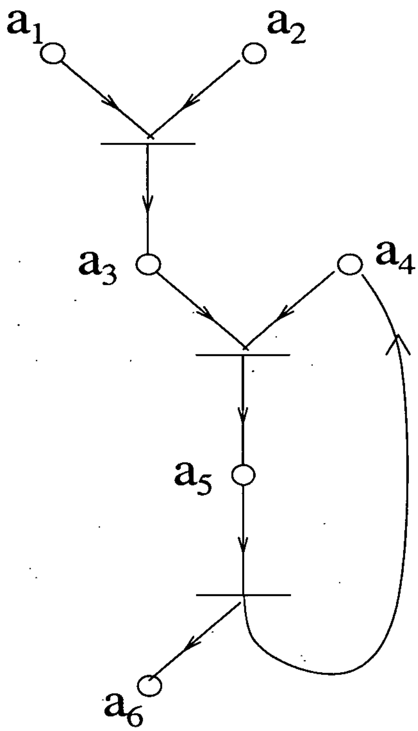

Figure 1. The process graph of Example 1.

For excluding the non-executability, we modify problem (1) in such a way that its feasible solutions will represent the executable feasible processes. For this reason, we use the following coloring of the process graphs. 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 procedure below.

Coloring Procedure

Step 1. Color all of the materials in $\bar{M} \cap R$.

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

Now, we can define the modified optimization problem. Let a process design problem $\mathbf{M} = (M, O, P, R)$ be given. A subgraph $(\bar{M}, \bar{O})$ of (M, O) is called a *feasible solution* of $(M, O, P, R,)$ if the following conditions are satisfied:

$(\bar{M}, \bar{O})$ satisfies $(A1)$ through $(A4)$,

moreover,

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

The set of the feasible solutions of $\mathbf{M}$ is denoted by $S'(M, O, P, R,)$ or $S'(\mathbf{M})$. Let w be a positive real-valued function defined on O. Then the modified optimization problem can be defined as follows.

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

Let us investigate the relationship between the feasible solutions of (2) and the executable feasible processes. First let us consider an executable feasible process. Obviously it determines a subgraph $(\bar{M}, \bar{O})$ of (M, O) uniquely. The following properties can be excepted from an executable feasible process.

Evidently, it must be executable. This yields that $(\bar{M}, \bar{O})$ is colorable by R.

It has to produce every desired products. This results in $P \subseteq \bar{M}$, *i.e.* $(\bar{M}, \bar{O})$ satisfies $(A1)$.

A material can be regarded as a raw material in the process if it is not to be produced by any available operating unit of the process under consideration. On the other hand, it can be excepted that all the materials other than the raw materials are to be produced by some operating unit of the process. This implies that $(A2)$ is valid for $(\bar{M}, \bar{O})$.

The appearance of an operating unit in the structurally feasible process is forbidden unless the corresponding operating unit participates directly or indirectly in the production of the desired products. This yields that $(A3)$ holds for $(\bar{M}, \bar{O})$.

Each of the materials of the process must be consumed or produced by at least one of the operating units of the process. This implies that $(\bar{M}.\bar{O})$ satisfies $(A4)$.

Summarizing we have that the P-graph $(\bar{M}, \bar{O})$, determined by the executable feasible process considered, satisfies conditions $(A1)$ through $(A4)$, moreover, it is colorable by R, and hence it is a feasible solution of (2).

Now, let us consider the reverse situation. Let $(\bar{M}, \bar{O}) \in S'(\mathbf{M})$. Let us consider the process based on $(\bar{M}, \bar{O})$ from structural point of view. Such a process exists and unique. Since $(\bar{M}, \bar{O})$ is a subgraph of (M, O), the process consists of only available operating units. Condition $(A1)$ ensures that all the desired products are produced in this process. Condition $(A2)$ guarantees that all the unproduced materials available in the process are raw materials. Conditions $(A3)$ and $(A4)$ imply that this process does not contain unnecessary operating units and unnecessary materials. Finally, the colorability of $(\bar{M}, \bar{O})$ provides that the process is executable. Indeed, since the process graph is colorable, we can assign the time to every operating unit when its output materials are colored. Then by choosing a suitable time unit, the coloring time of every operating unit can be considered as its scheduling time, and the process is executable. Of course this scheduling is not necessarily optimal. Therefore, this process is an executable feasible process.

Obviously, problem (2) is such a restriction of (1), where we are to find a minimum-weight feasible solution among the feasible solutions of (1) which represent executable processes. We note that problem (2) is also NP-hard. It can be proved in the same way as for (1) (*cf.* [1]). We also note that if the process graph (M, O) of $\mathbf{M} = (M, O, P, R)$ is cycle-free, then $S'(\mathbf{M}) = S(\mathbf{M})$, and therefore, Problems (1) and (2) collapse. Regarding the solution of cycle-free PNS problems, we refer to [2], [3], and [9].

To close this section we recall some notions on automata. By an *automaton* we mean a system $\mathbf{A} = (A, X)$, where A is a finite nonvoid set of *states*, X is a finite nonempty set of *input signs*, and every $x \in X$ is realized as a unary operation $x^{\mathbf{A}}$ on A. For any $a \in A$ and $x \in X$, $ax^{\mathbf{A}}$ can be interpreted as the state into which $\mathbf{A}$ enters from a by receiving the input sign x. For a word $p \in X^*$, $ap^{\mathbf{A}}$ can be defined inductively as follows:

(1) $a\varepsilon^{\mathbf{A}} = a$,

(2) $ap^{\mathbf{A}} = (av^{\mathbf{A}})x^{\mathbf{A}}$ for $p = vx$, $v \in X^*$ and $x \in X$,

where ε denotes the empty word of X^*.

One can assign a directed *transition graph* to each automaton as follows. Let $\mathbf{A} = (A, X)$ be an arbitrary automaton. By the *transition graph* of $\mathbf{A}$ we mean the graph $\mathcal{G}_{\mathbf{A}} = (A, E)$, where for any couple of states $a, b \in A$, $(a, b) \in E$ if and only if there exists an input sign $x \in X$ such that $ax^{\mathbf{A}} = b$. Let us equip each edge of the transition graph with a label which is equal to the corresponding letter as usual.

A *recognizer* is a system $\mathcal{A} = (\mathbf{A}, a_0, F)$ which consists of an automaton $\mathbf{A} = (A, X)$, the *initial state* $a_0 (\in A)$, and the set $F (\subseteq A)$ of *final states*. The language *recognized* by $\mathcal{A}$ is

$$L(\mathcal{A}) = \{p : p \in X^* \text{ and } a_0 p^{\mathbf{A}} \in F\}.$$

It is also said that $L(\mathcal{A})$ is *recognizable* by the automaton $\mathbf{A}$.

2 Solution of the modified PNS problem

We shall solve problem (2), by using an automaton theoretical approach. Namely, for every instance of (2), an automaton is constructed such that some feasible solutions of (2) can be described as words over the input alphabet of this automaton, moreover, these words are accepted by a recognizer based on this automaton. Then, by equipping the transition graph of the automaton considered with the weigths of the operating units, a shortest path in this weighted graph which leads from the initial state into the set of final states of this recognizer provides an optimal solution of (2).

We shall use the following statement.

Lemma. *Let an instance* $\mathbf{M} = (M, O, P, R)$ *of the process design problem be given. Moreover, let* $(\bar{M}, \bar{O})$ *be a process graph which is colorable by* R *and satisfies conditions* $(A1), (A2)$*, and* $(A4)$*, but* $(A3)$ *is not valid for* $(\bar{M}, \bar{O})$*. Then there exists a proper subgraph of* $(\bar{M}, \bar{O})$ *which is colorable by* R *and satisfies all the four conditions.*

Proof. We present a procedure to construct the required subgraph.

Procedure

Initialization Let $K_0 = P$, $O_0 = \emptyset$, and $i = 0$

Iteration (i-*th* iteration)

Step 1. Terminate if $K_i \subseteq R$; the required subgraph is $(mat(O_i), O_i)$. Otherwise proceed to Step 2.

Step 2. Select a material $x \in K_i \setminus R$ and an operating unit $u \in \bar{O}$ such that $x \in mat^{out}(u)$. Let $O_{i+1} = O_i \cup \{u\}$ and $K_{i+1} = (K_i \cup mat^{in}(u)) \setminus mat^{out}(O_{i+1})$. Set $i := i + 1$, and proceed to the succeeding iteration step.

The procedure is correct, since the colorability of $(\bar{M}, \bar{O})$ implies that if $K \not\subseteq R$ $(K \subseteq \bar{M})$, then there are $x \in K \setminus R$ and $u \in \bar{O}$ with $x \in mat^{out}(u)$. Now, let us suppose that the procedure is finished by the process graph (M_i, O_i), where $M_i = mat(O_i)$. Obviously, (M_i, O_i) is a subgraph of $(\bar{M}, \bar{O})$, moreover, $K_i \subseteq R$. These facts imply that (M_i, O_i) satisfies condition $(A2)$. From $M_i = mat(O_i)$ it follows that (M_i, O_i) satisfies $(A4)$. $K_0 = P$ implies that $(A1)$ is valid for (M_i, O_i). Finally, from the procedure it follows that $(A3)$ is also valid for (M_i, O_i), and therefore, (M_i, O_i) satisfies all the four conditions. Now, if (M_i, O_i) is not a proper subgraph of $(\bar{M}, \bar{O})$, then the two process graphs are equal. But this equality contradicts our assumption that $(\bar{M}, \bar{O})$ does not satisfy condition $(A3)$. Consequently, (M_i, O_i) is a proper subgraph of $(\bar{M}, \bar{O})$. Finally, it can be proved by induction on j that if each material of K_j has got color, then the materials contained in $mat(O_j)$ can be colored by K_j. From this fact it follows that (M_i, O_i) can be colored by R, which ends the proof of the statement.

To construct the automaton mentioned above, let us consider an arbitrary instance $\mathbf{M} = (M, O, P, R)$ of the design problem and let w be a weight function.

Let us define the automaton $\mathbf{B} = (B, O')$ as follows. Let $B = B' \cup \{\diamond\}$, where $B' = \wp'(M)$ and $\diamond \notin B'$. Moreover, let $O' = \{u : u = (C, D) \in O \text{ and } R \cap D = \emptyset\}$. One can give the states of the automaton the following meaning. A state which is a set of materials means the available materials at a given time. State $\diamond$ is used for describing the unsuccessful transitions. The transitions are defined in the following way. For every $Q \in B'$ and $u = (C, D) \in O'$, let

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

The transitions have the following meaning. Let us suppose that we are going to build up a process graph. First we fix the available materials, their set Q will be the starting state of the automaton. As a next step, we try to put an operating unit in the graph, let $u = (C, D)$ denote it. If each input material of u is available at this moment, *i.e.*, $C \subseteq Q$, then we can put u in the process graph, and we can suppose that from this moment the available materials are the earlier available ones and the output materials of u, *i.e.*, the elements of the set $Q \cup D$. If u has such an input material which is not available, then we can not put u in the process graph we build, and this fact is expressed such that the transition is unsuccessful. It is easy to check that the following observation is valid for the automaton **B**.

Remark 2. If Q is a state of $\mathbf{B}$, p is a word over O', and $u \in O'$ occurs in p, then $Q(pu)^{\mathbf{B}} = Qp^{\mathbf{B}}$.

Let us equip the transition graph $\mathcal{G}_{\mathbf{B}}$ with weights in the following way. If (Q, Q') is an edge of $\mathcal{G}$ and the labels of this edge are $u_{j_1}, \ldots, u_{j_t}$, then let us assign the weight $w' = \min\{w(u_{j_1}), \ldots, w(u_{j_t})\}$ to the edge under consideration, moreover, if $w' = u_{j_l}$ for some $1 \leq l \leq t$, then keep the label u_{j_l} and cancel the remaining labels of this edge. Let us denote this weighted and labelled graph by $(\mathcal{G}_{\mathbf{B}}, w)$.

Let us define now the recognizer $\mathcal{B} = (\mathbf{B}, R, F)$, where $F = \{Q : Q \subseteq B' \text{ and } P \subseteq Q\}$. Then the following statement is valid.

Proposition. *For every word $p = u_{i_1} \ldots u_{i_k} \in L(\mathcal{B})$, if $path[R, Rp^{\mathbf{B}}]$ is a shortest path among the paths leading from R into a final state in $(\mathcal{G}_{\mathbf{B}}, w)$, then $u_{i_1}, \ldots, u_{i_k}$ are pairwise different and $(\bar{M}, \bar{O})$ is an optimal solution of (2), where $\bar{O} = \{u_{i_1}, \ldots, u_{i_k}\}$ and $\bar{M} = mat(\bar{O})$.*

Proof. Let $p = u_{i_1} \ldots u_{i_k} \in L(\mathcal{B})$ and let us suppose that $path[R, Rp^{\mathbf{B}}]$ is a shortest path leading from R into a final state in $(\mathcal{G}_{\mathbf{B}}, w)$. Then Remark 2 implies that $u_{i_1}, \ldots, u_{i_k}$ are pairwise different, since every operating unit has a positive weight. Now, let us consider the process graph $(\bar{M}, \bar{O})$, where $\bar{O} = \{u_{i_1}, \ldots, u_{i_k}\}$ and $\bar{M} = mat(\bar{O})$. First we show that $(\bar{M}, \bar{O})$ is a feasible solution of (2). From the definition of $(\bar{M}, \bar{O})$ it follows that $(A4)$ holds for $(\bar{M}, \bar{O})$. The definition of O' and $p \in L(\mathcal{B})$ imply that $(A2)$ is valid for $(\bar{M}, \bar{O})$. Moreover, from $p \in L(\mathcal{B})$ it follows that $(\bar{M}, \bar{O})$ is colorable by R and $(A1)$ is valid for $(\bar{M}, \bar{O})$. It is stated now that $(\bar{M}, \bar{O})$ satisfies $(A3)$. If it is not so, then by our Lemma, there exists a proper subgraph of $(\bar{M}, \bar{O})$ which is a feasible solution of (2). Let us denote this subgraph by $(\widehat{M}, \widehat{O})$. Since $\bar{M} = mat(\bar{O})$ and by Remark 1, $\widehat{M} = mat(\widehat{O})$, we obtain that $\widehat{O} \subset \bar{O}$. Let us suppose that $\widehat{O} = \{u_{j_1}, \ldots, u_{j_l}\} \subset \bar{O}$ for some $1 \leq l < k$. Since $(\widehat{M}, \widehat{O})$ is a feasible solution of (2), it is colorable by R. Without loss of generality, we may assume that the coloring procedure first colors the output materials of u_{j_1}, then the output materials of u_{j_2} *etc.* This yields that the word $\hat{p} = u_{j_1} \ldots u_{j_l}$ brings the automaton $\mathbf{B}$ from R into some final state. On the other hand, the weight of $path[R, R\hat{p}^{\mathbf{B}}]$ is less than the weight of $path[R, Rp^{\mathbf{B}}]$ which is a contradiction. Therefore, $(\bar{M}, \bar{O})$ satisfies $(A3)$, and it is a feasible solution of (2). Let us observe that the weight $\bar{w}$ of $(\bar{M}, \bar{O})$ is equal to the weight of $path[R, Rp^{\mathbf{B}}]$. Now, we prove that $(\bar{M}, \bar{O})$ is an optimal solution of (2). Indeed, if it is not so, then there exists a feasible solution $(\widehat{M}, \widehat{O})$ of (2) such that its weight $\hat{w}$ is less than the weight $\bar{w}$ of $(\bar{M}, \bar{O})$. In similar way as above, we can construct then a word $\hat{p}$ such that $\hat{p} \in L(\mathbf{B})$ and $\hat{w}$ is equal to the weight of $path[R, R\hat{p}^{\mathbf{B}}]$. This yields that the weight of $path[R, R\hat{p}^{\mathbf{B}}]$ is less than the weight of $path[R, Rp^{\mathbf{B}}]$ which contradicts our assumption that $path[R, Rp^{\mathbf{B}}]$ is a shortest path leading from R into some final state in $(\mathcal{G}_{\mathbf{B}}, w)$. Consequently, $(\bar{M}, \bar{O})$ is an optimal solution of (2).

Our Proposition provides the following procedure for finding an optimal solution of (2).

Procedure for finding an optimal solution of (2)

Step 1. Construct the transition graph of the automaton $\mathbf{B}$ and calculate the set F of final states.

Step 2. Let us equip the transition graph with the weights of the operating units, and simultaneously, rewrite the labels of the edges such that let every edge have only one label.

Step 3. Determine a shortest path leading from the state R into the set F.

Step 4. By using the obtained shortest path, determine an optimal solution of problem (2).

It is worth noting that the whole transition graph is not required by the procedure in general, only the transition graph of the subautomaton generated by the state R. To demonstrate this fact and the procedure, let us consider the following small example.

Example 2. Let $M = \{a_1, \ldots, a_8\}$, $R = \{a_1, a_2, a_3\}$, $P = \{a_8\}$ and $O = \{u_1, u_2, u_3, u_4\}$, where the definition of the operating units and their weights are given by the table below.

	input materials	output materials	weight
u_1	a_1, a_2	a_4, a_5	2
u_2	a_2	a_5, a_6	5
u_3	a_1, a_3	a_6, a_7	1
u_4	a_5, a_6	a_8	3

The process graph of this design problem is depicted in Figure 2.

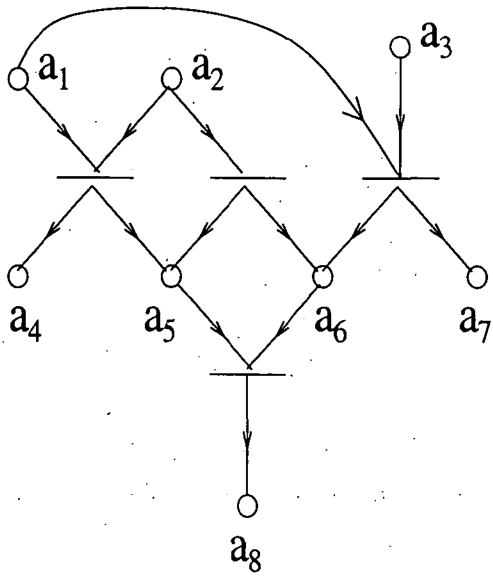

Figure 2. The process graph of Example 2.

By constructing the transition graph of the subautomaton generated by R, we get a transition graph of 12 vertices. It is depicted in Figure 3, where the sets are given by circles containing the indices of their elements, loop edges are omitted, furthermore, over each edge the operating unit is written which induces the transition and under the edge the weight of the operating unit is given. By determining the shortest paths, we obtain that the path belonging to the word $u_1u_3u_4$ is a suitable shortest path, its edges are bold in Figure 3. The corresponding optimal solution with weight 6 is given in Figure 4.

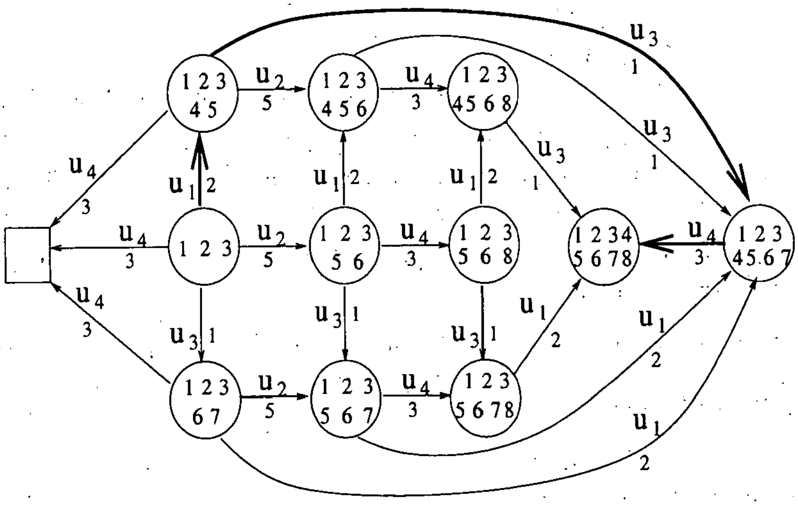

Figure 3. The weighted transition graph for Example 2.

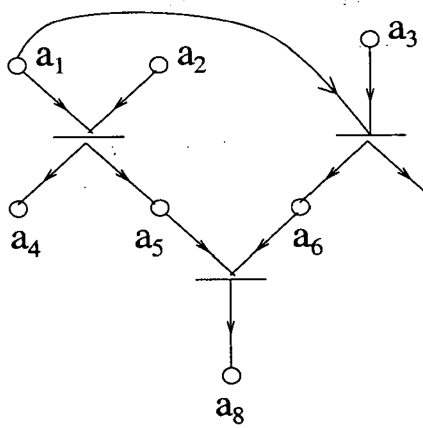

Figure 4. The optimal solution of Example 2.

References

[1] Blázsik, Z. and 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, Cs. Imreh, Z. Kovács, On a well-solvable class of the PNS problem, *Novi Sad Journal of Mathematics* **30** (2000), 21-30.

[3] Blázsik, Z., Cs. Holló, Cs. Imreh, Z. Kovács, Heuristics for the PNS Problem, *Mátraháza Optimization Days*, editors: F. Gianessi, P. Pardalos, T. Rapcsák, Kluwer Academic Publisher, to appear.

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

[5] Friedler, F., 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.

[6] Friedler, F., K. Tarján, Y.W. Huang, and L.T. Fan, Combinatorial Algorithms for Process Synthesis, *Computers Chem. Engng.* **16** (1992), S313-S320.

[7] 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*, (editors: I. M. Bonze, T. Csendes, R. Horst, P. M. Pardalos), Kluwer Academic Publisher, Dordrecht, Boston, London, 1996, 301-348.

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

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

Received March, 2001