

Control of Deadlocked Discrete-Event Systems Using Petri Nets

František Čapkovič

Institute of Informatics, Slovak Academy of Sciences, Dúbravská cesta 9, 845 07 Bratislava, Slovakia, e-mail: Frantisek.Capkovic@savba.sk

Abstract: In Discrete-Event Systems (DES), deadlocks frequently occur. Flexible Manufacturing Systems (FMS) have the character of DES. Namely, FMS consist of many cooperating devices (like robots, machine tools, transport belts, etc.). Frequently, deadlocks occur because of insufficient resources. Petri Nets (PN) are often used to model FMS and to synthesize control for them. To deal with deadlocks, first of all, it is necessary to find and/or avoid them. There are several principal approaches for doing this - either by computing and analyzing the PN reachability tree (RT) or by finding PN model siphons. Then, in the former concept, the supervisor is synthesized by means of P-invariants of the PN model used, while in the latter concept the supervisor, based on siphons, is synthesized. In addition to these approaches, additional techniques can sometimes, be applied - e.g. a suitable utilization of added PN transitions.

Keywords: control; deadlocks; discrete-event systems; Petri nets; P-invariants; siphons

1 Introduction

In Discrete-Event Systems (DES) a next state depends only on the actual state and on the occurrence of discrete events. For modeling and control of DES Petri nets (PN) are frequently used. One of the typical representatives of DES is the family of Flexible Manufacturing Systems (FMS), newer Automated Manufacturing Systems (AMS). In such systems (robotized working cells, discrete production lines, and the like) many devices cooperate together - robots, machine tools, transport belts, automatically guided vehicles (AGV), etc. They are called to be *resources*. Inside FMS/AMS the resource allocation is very important. Hence, Resource Allocation Systems (RAS) are investigated.

For above mentioned reasons *deadlocks* often occur in RAS. Deadlocks are, of course, undesirable and unfavorable. They disrupt the normal course of the production process. Due to deadlocks, it remains stagnate. Thus, the primary intention of the production cannot be achieved. Deadlocks can arise, for example, when a machine M, completes a part and there is no part in a buffer to be fed to M,

it is the situation called *starvation*. In general, release of too few parts to RAS may starve some machines and lower their production rate. Therefore, it is necessary to pursue the maximally permissive control policy, for deadlock avoidance [1] by releasing as many jobs as possible into the system. On the other hand, when a machine M completes a part that cannot be unloaded because of the lack of buffer spaces, it is *blocked*. In general, *blocking* is caused by the excessive job releases and limited buffer spaces. Blocked machines are forced to be idle, thereby they lose their productivity. The more parts occur in the system, the more likely is occurrence of deadlocks and the machines are blocked. To operate RAS effectively, the system should be well scheduled and deadlocks should be completely avoided, in this way, the reduction of starvation and blocking is efficiently achieved.

To deal with the deadlocks, it is necessary to find them, and to find a suitable methodology for how to eliminate their impact and to successfully control the system. The deadlocks in FMS can be found by applying two main manners:

- (i) Finding and analyzing the reachability tree (RT) of the PN model representing the causality of PN states.
- (ii) Finding and analyzing structural properties - namely the set of siphons and traps of the PN model. Traps are some complements of siphons. Sometimes, the approach using an application of additional transition(s) into the PN model may be very suitable.

As to the control of deadlocked FMS, i.e. elimination of the deadlock impact, three concepts of synthesizing the supervisor are used here:

- (i) The approach based on P -invariants of the PN model and simultaneous utilizing its RT
- (ii) The approach-based on PN model siphons and traps
- (iii) The auxiliary approach adding some supplementary transitions to the PN model

1.1 The State of the Art Review

Deadlocks are looking for and analyzed for tens of years in software engineering and other branches, and for a long time also in FMS/AMS.

Among pioneers of deadlock avoidance in DES and RAS belongs S. A. Reveliotis with his school. Their oldest publications were devoted especially to software engineering, but their newer ones - see [2-4] - are concerning RAS in FMS/AMS. Another school around P. J. Antsaklis - see [5] - is specialized on RT-based approaches to the deadlock avoidance. Both schools are American.

There are many other authors, even schools, interested in this area. The newer schools are the Chinese-American school around Meng Chu Zhou and Zhi Wu Li -

see [6-9] specialized on siphon-based approaches. This school publishes very intensively. Smaller schools are the French school around K. Barkaoui - see e.g. [10], and several others schools - in Spain [12], Italy [12-14], Germany [15].

But at present the peak school is the pure Chinese school with enormous number of authors from various universities - see e.g. [16-18] - with top results. Some authors from the above introduced schools publish also together cross by cross the schools. It is impossible to make a complete overview of all the works of these schools on a limited number of pages per paper.

Deadlocks may occur also in DES and RAS with non-determinism analysed in the paper [19]. Such deadlocks have to be avoided too.

Simply, this area of research in FMS/AMS lives through a boom. Therefore, it is useful to choose the more important approaches and compare them. The best form is to do this by applying them on the same real plant and compare and evaluate their results. Such a process has not been published until now.

1.2 The Main Aim of the Paper

The main aim of this paper is to point out:

- (i) The three principal kinds of approaches extracted from the huge amount of literature
- (ii) How to avoid deadlocks in real RAS
- (iii) How to synthesize the control of RAS by means of PN models
- (iv) How to apply particular approaches on the same real discrete plants in order to compare them
- (v) How to perform the comparison on the basis of achieved results and how they are evaluated. (In the literature, the author of this paper did not find such comparison and evaluation of different approaches).

Of course, finding the computational complexity of algorithms for computing RT (at the state analysis) and minimal siphons (at the structural analysis), respectively, is also an associated, but not less important, aim.

For FMS/AMS practice such a comparison and evaluation may be very useful. Namely, on one hand it is important to avoid deadlocks, but on the other hand it is also necessary to detect whether the proposed supervisor avoiding deadlocks ensures satisfying functionality of RAS or not. If not, a structural reconstruction of the original system and/or changing the number of resources and repeating the procedure of the supervisor synthesis is needful.

1.3 Paper Organization

Here, in this Section 1, the state of the art review and the main aim of this paper are introduced. In the Section 2, the PN themselves as well as PN P -invariants and PN siphons and traps were defined and two approaches how to control RAS were sketched. In the next Section 3 the auxiliary simple approach to control of deadlocked RAS will be introduced and illustrated on an example, namely the approach based on additional (supplementary) transitions. Next, in the Section 4, the second approach to solving that problem will be introduced and illustrated on examples, namely the approach based on P -invariants. In the Section 5 the approach based on siphons and traps will be presented and illustrated on examples. In the Section 6 both approaches will be compared (as to their advantageous and disadvantageous) and evaluated. In the Conclusions, the final view on the dealing with deadlocks in this paper and the plans for future research will be introduced.

2 Preliminaries

PN are perspective tool [12] for modeling and DES. Essentials of PN were presented in many older papers - see e.g. [10]. The state equation of place/transition PN (P/T PN) - see [19] - is the following

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{B} \cdot \mathbf{u}_k, k = 0, 1, \dots \quad (1)$$

$$\mathbf{F} \cdot \mathbf{u}_k \leq \mathbf{x}_k \quad (2)$$

where

$\mathbf{x}_k = (x_{p_1}, x_{p_2}, \dots, x_{p_n})^T$ is the state vector with integer entries $x_{p_i} \in \{0, 1, \dots, \infty\}$ being the states of particular places p_i , $i = 1, \dots, n$, in the step k , namely x_{p_i} represents the actual number of tokens in the place p_i . The vector \mathbf{x}_0 is the initial state vector.

$\mathbf{u}_k = (u_{t_1}, u_{t_2}, \dots, u_{t_m})^T$ is the control vector with entries $u_{t_j} \in \{0, 1\}$ being the states of particular transitions t_j , $j = 1, \dots, m$, in the step k . They can be disabled or enabled. The disabled t_j cannot be fired, i.e. $u_{t_j} = 0$, while enabled t_j may be (but needs not be) fired i.e. $u_{t_j} = 1$. In P/T PN enabled transitions represent the occurrence of discrete events.

$\mathbf{B} = \mathbf{G}^T - \mathbf{F}$ is the structural matrix of integers with \mathbf{G} being the incidence matrix of directed arcs from transitions to places while \mathbf{F} being the incidence matrix of directed arcs from places to transitions

Let $P = \{p_1, \dots, p_n\}$ and $T = \{t_1, \dots, t_m\}$ are, respectively, the set of PN places and the set of PN transitions. Thus, $\mathbf{F} = \{f_{ij}\}$, $i = 1, \dots, n$; $j = 1, \dots, m$, $f_{ij} \in Z$, where Z is the set of integers, and it represents the existence and multiplicity of arcs directed

from p_i to t_j ; $\mathbf{G} = \{g_{ji}\}, j = 1, \dots, m; i = 1, \dots, n, g_{ji} \in \mathbb{Z}$, and it represents the existence and multiplicity of arcs directed from t_j to p_i .

\mathbf{x}_0 is the initial state vector.

Starting from \mathbf{x}_0 and firing an enabled transition the next state \mathbf{x}_1 can be reached. The reachability tree (RT) expresses all possible branches of the development of the system (1), (2). A firing sequence of transitions t_a, t_b, \dots, t_c represents a branch $\mathbf{x}_0 \xrightarrow{u_{t_a}} \mathbf{x}_1 \xrightarrow{u_{t_b}} \dots \mathbf{x}_{k-1} \xrightarrow{u_{t_c}} \mathbf{x}_k$ of RT. All reachable states create the state space, i.e. the set $\mathcal{R} = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k\}$. PN transitions symbolize edges of RT. By means of the thorough analysis of RT (either in graphical form or in the form of the adjacency matrix) all deadlocks can be found. Then the supervisor based on P -invariants can be synthesized. The P -invariant is the $(n \times 1)$ -dimensional vector $\mathbf{y} \neq \mathbf{0}$ for which $\mathbf{y}^T \cdot \mathbf{B} = \mathbf{0}$.

A nonempty subset $S \subset P$ in P/T PN is called a *siphon* if every transition having an output place in S has an input place in S . A nonempty subset $Q \subset P$ in P/T PN is called a *trap* if every transition having an input place in Q has an output place in Q . Siphons create a set of places which, if become empty of tokens, will always *remain empty* for all reachable markings of the net. When all places in a siphon have no token, all transitions connecting with the siphon can no longer be fireable. Traps create a set of places which, if become marked, will always *remain marked* for all reachable markings of the net. The union of two siphons (traps) is again a siphon (trap).

If every non-empty siphon of PN includes a sufficiently marked trap then - see e.g. [6], no dead marking is reachable. It is very important piece of knowledge. Thorough analysis of siphons and traps is a path to the proposal of the supervisor. Then, the supervisor will be synthesized by means of utilizing properties of siphons and traps.

It is not necessary to work with all siphons (there are many). It is sufficient to work with *elementary siphons*, i.e. linearly independent siphons. Even, it is sufficient to work with *minimal siphons* and *minimal traps* - see e.g. [11] [20].

In next, both approaches to control of deadlocked FMS, especially of the special kind of FMS called RAS, will be presented. Namely, above mentioned devices - machine tools, robots, buffers, transport belts, AGV, and so on, can be understood to be various resources. The resources are usually shared by two or more subsystems of RAS. Because of a limited number of resources different kinds of problems, especially deadlocks, arise during the system operation.

3 Approach Based on Additional Transitions

This approach is very useful especially in the case of so called *diamonds* in RT of the deadlocked PN model. The diamond $\hat{\sigma}$ from the start state \mathbf{x}_a in RT to the end state \mathbf{x}_b , $\mathbf{x}_a \neq \mathbf{x}_b$, is a pair of paths $\hat{\sigma} = \langle \mathbf{x}_a \sigma_1 \mathbf{x}_b, \mathbf{x}_a \sigma_2 \mathbf{x}_b \rangle$, where paths $\sigma_1 \cap \sigma_2 = \emptyset$, $\sigma_1 \cup \sigma_2 \neq \emptyset$, with \emptyset being the empty set, and $\mathbf{x}_a, \mathbf{x}_b$ do not belong in $\sigma_1 \cup \sigma_2$. When \mathbf{x}_b is the deadlock, the following approach is possible in order to deal with it. After adding a transition into the PN model, the structural matrix of the supervisor:

$$\mathbf{B}_c = \mathbf{x}_c - \mathbf{x}_b \quad (3)$$

where usually $\mathbf{x}_c = \mathbf{x}_0$ (being the initial state of the PN model (1)-(2)).

3.1 Example 1

Consider the RAS in Figure 1 consisting of three loading buffers I1–I3 and three unloading buffers O1–O3. They, respectively, load and unload the FMS corresponding to three raw product types, Pr1–Pr3, to be processed by machine M. They are moved by robot R. The production cycles are the following: a raw product Pr1 is taken from I1 by R and put in M. After being processed by M, it is taken by R and put to output O1. A raw product Pr2 is taken by R from I2, processed by M and then moved by R from M to O2. A raw product Pr3 is taken by R from I3, processed by M and moved by R from M to O3. The PN model is given in Fig. 2. The initial state $\mathbf{x}_0 = (8 \ 0 \ 0 \ 8 \ 0 \ 0 \ 1 \ 0 \ 4 \ 4)^T$. Because figures of RT from \mathbf{x}_0 produced by a graphic tool has a poor quality at a greater dimensionality of RT, the deadlocks will be computed by means of the zero rows of the RT adjacency matrix. There are 32 nodes (states) in the RT.

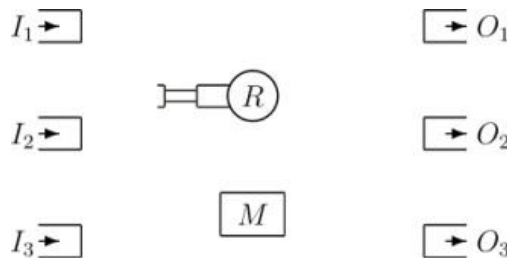


Figure 1
The scheme of RAS

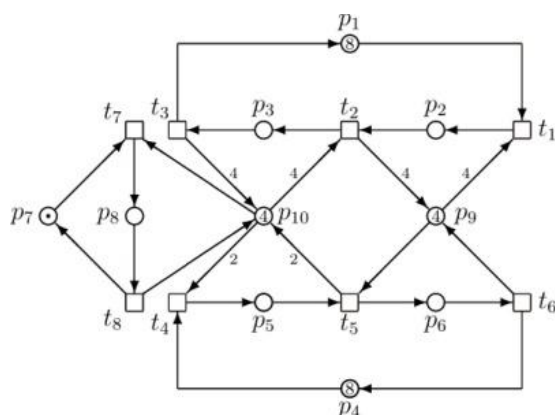


Figure 2

The PN model of the deadlocked RAS

The structural matrix of the PN model and its initial state are the following:

$$\mathbf{B} = \begin{pmatrix}
 -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 \\
 -4 & 4 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\
 0 & -4 & 4 & -2 & 2 & 0 & -1 & 1 & 1
 \end{pmatrix}$$

$$\mathbf{x}_0 = (8 \ 0 \ 0 \ 8 \ 0 \ 0 \ 1 \ 0 \ 4 \ 4)^T$$

Small numbers in the neighborhood of some directed arcs mean their multiplicity.

This plant has only one deadlock – the state $\mathbf{x}_{11} = (7 \ 1 \ 0 \ 6 \ 2 \ 0 \ 1 \ 0 \ 0 \ 0)^T$ (i.e. numbers of tokens in corresponding places are: $p_1 = 7, p_2 = 1, p_4 = 6, p_5 = 2, p_7 = 1$) is the deadlock. Thus, $\mathbf{B}_c = \mathbf{x}_0 - \mathbf{x}_{11} = (1 \ -1 \ 0 \ 2 \ -2 \ 0 \ 0 \ 4 \ 4)^T$, where the vector $\mathbf{B}_c^{(-)} = (0 \ 1 \ 0 \ 0 \ 2 \ 0 \ 0 \ 0 \ 0)^T$ represents the multiplicity of directed arcs from corresponding PN places to the added transition and the vector $\mathbf{B}_c^{(+)} = (1 \ 0 \ 0 \ 2 \ 0 \ 0 \ 0 \ 0 \ 4 \ 4)^T$ represents those from the added transition to corresponding PN places. The PN model of the modified structure is given in Figure 3. Alike as in previous case, the RT is not introduced in the graphical form. From the RT adjacency matrix, it follows that PN has 32 nodes and no deadlock exists in the supervised system displayed in Fig. 3. Adding the transition t_9 into the PN model of RAS given in Figure 2 and applying its interconnections with the original model through the relation (3) the deadlock \mathbf{x}_{11} was eliminated.

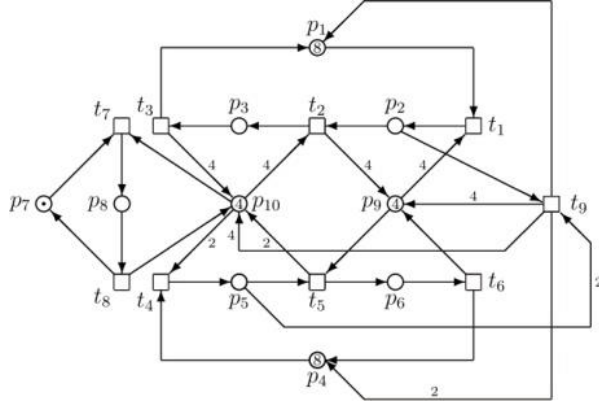


Figure 3

The PN model of RAS supervised by means of the transition t_9 without any deadlock

4 Approach Based on P-invariants

As it was mentioned above a vector $\mathbf{y} \neq \mathbf{0}$ fulfilling the relation $\mathbf{y}^T \cdot \mathbf{B} = \mathbf{0}$ is named as the invariant. For more invariants, e.g. s , the matrix \mathbf{Y} consisting of s invariants being its columns, has to fulfill:

$$\mathbf{Y}^T \cdot \mathbf{B} = \mathbf{0} \quad (4)$$

Putting a restricted condition:

$$\mathbf{L} \cdot \mathbf{x} \leq \mathbf{b}, \quad (5)$$

where \mathbf{L} is a $(s \times n)$ matrix of positive integers, expressing by its rows suitable linear combinations of state vectors entries (i.e. numbers of tokens inside corresponding PN places), and \mathbf{b} is a $(s \times 1)$ column vector of limits for each row of \mathbf{L} (i.e. a maximal number of tokens in places in the corresponding row together). To remove inequality in (4), we can put the following:

$$\mathbf{L} \cdot \mathbf{x} + \mathbf{x}_s = \mathbf{L} \cdot \mathbf{x} + \mathbf{I}_s \cdot \mathbf{x}_s = (\mathbf{L} \ \mathbf{I}_s) \cdot (\mathbf{x}^T \ \mathbf{x}_s^T)^T = \mathbf{b} \quad (6)$$

where \mathbf{I}_s is the $(s \times s)$ identity matrix. To synthesize the supervisor with the structure \mathbf{B}_s (unknown till now), we force $(\mathbf{L} \ \mathbf{I}_s)$ into (4) instead of \mathbf{Y}^T as well as $(\mathbf{B}^T \ \mathbf{B}_s^T)^T$ instead of \mathbf{B} . In such a way we finally obtain the supervisor structure:

$$\mathbf{B}_s = -\mathbf{L} \cdot \mathbf{B} \quad (7)$$

$$\mathbf{B}_s = \mathbf{G}_s^T - \mathbf{F}_s \quad (8)$$

$$\mathbf{F}_s = -\mathbf{B}_s^{(-)}; \quad \mathbf{G}_s^T = \mathbf{B}_s^{(+)}$$

$$\mathbf{x}_{0s} = \mathbf{b} - \mathbf{L} \cdot \mathbf{x}_0 \quad (9)$$

Besides (5) the *general linear constraints* can be imposed to be satisfied by the supervised system:

$$\mathbf{L}_p \cdot \mathbf{x} + \mathbf{L}_t \cdot \mathbf{u} + \mathbf{L}_v \cdot \mathbf{v} \leq \mathbf{b} \quad (10)$$

where,

\mathbf{b} is s -dimensional nonnegative integer vector expressing some limits

\mathbf{L}_p , \mathbf{L}_t , \mathbf{L}_v are, respectively, $(s \times n)$ -, $(s \times m)$ -, $(s \times m)$ -dimensional matrices of integers. They concern, respectively, PN places, PN transitions and the Parikh's vector \mathbf{v} . The sense of the Parikh's vector \mathbf{v} is clear from the following relation expressing the evaluation of PN model (1), (2), i.e.

$$\mathbf{x}_k = \mathbf{x}_0 + \mathbf{B} \cdot (\mathbf{u}_0 + \mathbf{u}_1 + \dots + \mathbf{u}_{k-1}) = \mathbf{x}_0 + \mathbf{B} \cdot \mathbf{v} \quad (11)$$

As to (11), it was proved in [5] that when $\mathbf{L}_p \cdot \mathbf{x} - \mathbf{b} \leq \mathbf{0}$ the supervisor with the following structure and initial state

$$\mathbf{F}_s = \max(\mathbf{0}, \mathbf{L}_p \cdot \mathbf{B} + \mathbf{L}_v, \mathbf{L}_t) \quad (12)$$

$$\mathbf{G}_s^T = \max(\mathbf{0}, \mathbf{L}_t - \max(\mathbf{0}, \mathbf{L}_p \cdot \mathbf{B} + \mathbf{L}_v)) - \min(\mathbf{0}, \mathbf{L}_p \cdot \mathbf{B} + \mathbf{L}_v) \quad (13)$$

$$\mathbf{x}_{0s} = \mathbf{b} - \mathbf{L}_p \cdot \mathbf{x}_0 - \mathbf{L}_v \cdot \mathbf{v}_0 \quad (14)$$

guarantees that constraints are verified for the states resulting from the initial state \mathbf{x}_0 . Here, the $\max(\cdot)$ is the maximum operator of operands. For matrices it is applied element by element, i.e. $\mathbf{Z} = \max(\mathbf{X}, \mathbf{Y})$ means that $z_{ij} = \max(x_{ij}, y_{ij})$.

\mathbf{v}_0 is the $(m \times 1)$ vector containing nonzero entries (namely equal to 1) solely in positions of transitions being fireable in \mathbf{x}_0 .

Now, consider the RAS in Figure 1 with PN model given in Figure 2. Let us deal with the deadlock state \mathbf{x}_{11} using the P -invariants based approach for both versions the simpler (5) and the generalized (10).

4.1 Simpler Version of the Approach

Analyzing RT by means of the adjacency matrix we can reveal that the deadlocked state \mathbf{x}_{11} is reached by two ways:

$$\mathbf{x}_0 \xrightarrow{t1} \mathbf{x}_1 \xrightarrow{t4} \mathbf{x}_5 \xrightarrow{t4} \mathbf{x}_{11} \quad (15)$$

$$\mathbf{x}_0 \xrightarrow{t4} \mathbf{x}_2 \xrightarrow{t4} \mathbf{x}_7 \xrightarrow{t1} \mathbf{x}_{11} \quad (16)$$

The corresponding critical RT nodes are the following:

$$\mathbf{x}_5 = (7 \ 1 \ 0 \ 7 \ 1 \ 0 \ 1 \ 0 \ 0 \ 2)^T$$

$$\mathbf{x}_7 = (8 \ 0 \ 0 \ 6 \ 2 \ 0 \ 1 \ 0 \ 4 \ 0)^T$$

$$\mathbf{x}_{11} = (7 \ 1 \ 0 \ 6 \ 2 \ 0 \ 1 \ 0 \ 0 \ 0)^T$$

$$\mathbf{x}_5 - \mathbf{x}_{11} = (0 \ 0 \ 0 \ 1 \ -1 \ 0 \ 0 \ 0 \ 0 \ 2)^T$$

$$\mathbf{x}_7 - \mathbf{x}_{11} = (1 \ -1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 4 \ 0)^T$$

Using the approach (6)-(8) based on P -invariants, we have to put some restrictions on combinations of some critical places. Such places are p_2 , p_5 and moreover p_8 , p_{10} . Namely, the following restrictions have to be imposed:

$$p_8 + p_{10} \leq 5 \quad (17)$$

$$p_2 + p_5 \leq 1 \quad (18)$$

Consequently,

$$\mathbf{L} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (19)$$

$$\mathbf{b} = \begin{pmatrix} 5 \\ 1 \end{pmatrix} \quad (20)$$

Applying these matrices into (6)-(8) we obtain the supervisor structure and initial state as follows:

$$\mathbf{B}_s = \begin{pmatrix} 0 & 4 & -4 & 2 & -2 & 0 & 0 & 0 \\ -1 & 1 & 0 & -1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{F}_s = -\mathbf{B}_s^{(-)} = \begin{pmatrix} 0 & 0 & 4 & 0 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{G}_s^T = \mathbf{B}_s^{(+)} = \begin{pmatrix} 0 & 4 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{x}_{0s} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

The PN model of RAS controlled by such supervisor is displayed in Figure 4. The supervisor ensures that no deadlock occurs here. RT has 24 nodes in this case.

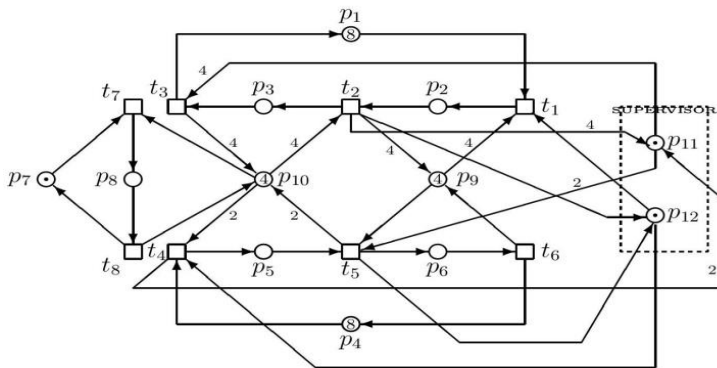


Figure 4

The PN model of RAS controlled by the P -invariants based supervisor removing the deadlock

After deeper analysis, we can found that the condition (17) may be omitted and it is sufficient in this RAS to use solely the condition (18). Thus, we obtain:

$$\mathbf{L} = (0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0)$$

$$\mathbf{b} = (1)$$

consequently,

$$\mathbf{F}_s = (1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0)$$

$$\mathbf{G}_{s}^T = (0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0)$$

$$\mathbf{x}_{0s} = (1)$$

The PN model of the controlled RAS is almost-certain as that on Figure 4, only place p_{11} together with its interconnections are missing. However, the RT is the same. It means that no deadlocks are indicated in it.

4.2 The Generalized Version of the Approach

Here we use the approach (10)-(14) to illustrate its potency. From the PN model of the original uncontrolled system in Figure 2 it can be seen that main problem consists in the places p_2 and p_5 as well as in Parikh's vectors v_1 and v_4 . Let us put $p_2 + p_5 \leq 1$ and because of the Parikh's vectors put $t_4 > t_1$ and $t_1 > t_4$. Hence:

$$\mathbf{L}_p = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}; \quad \mathbf{b} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\mathbf{L}_v = \begin{pmatrix} -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{pmatrix}; \quad \mathbf{L}_t = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$\mathbf{v}_0 = (1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0)^T$ is the initial vector expressing enabled transitions at the initial state \mathbf{x}_0 .

These inputs into (12) - (14) result the following synthesized supervisor

$$\mathbf{F}_s = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{G}_{s}^T = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{x}_{0s} = (1 \ 1)^T$$

The PN model of the controlled RAS is displayed in Figure 5 and its RT is given in Figure 6. For such a small RT like this, as opposed to greater ones, the graphical output of RT has a better quality, is readable and sufficiently highlighted. No deadlock occurs there.

As to evolution of the PN model behavior, restrictions on the supervised system are more rigorous. Thus, the RT is not so much branching out but in spite of this it guarantees a deadlock-free behavior of the PN model. It has only 13 nodes

(including the initial node) - states of the system. In comparison with the previous one given in Figure 4, having 24 nodes, it can be seen that this structure of controlled RAS yields only about half number of states, what may be insufficient from practical point of view (e.g. the functionality of a real plant, its utility, etc.). The user in practice has to consider if this structure is adequate for his requirements, or he will use the previous structure displayed in Figure 4. Namely, the too severe supervisor can hamper the required behavior of the system. In such a case the user has to change the structure of RAS and/or the number of resources and to repeat the whole process of the supervisor synthesis.

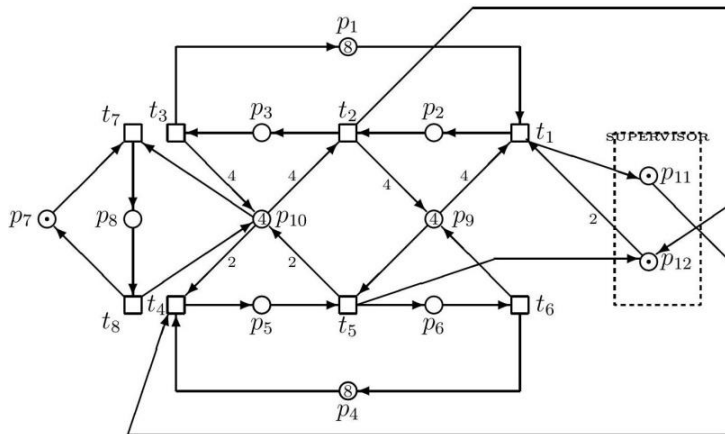


Figure 5

The PN model of the controlled RAS

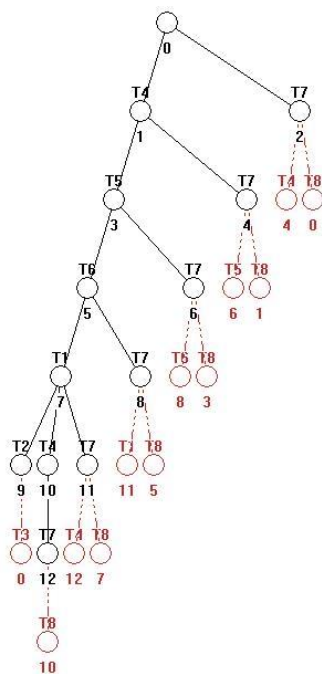


Figure 6
The RT of the PN model of the controlled RAS

4.3 Example 2

Consider RAS schematically sketched in Figure 7.

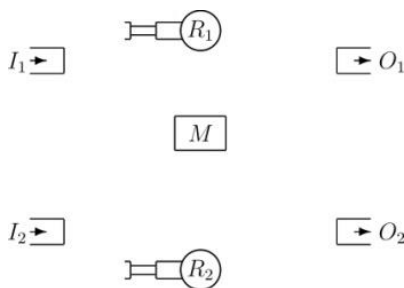


Figure 7
The scheme of RAS

There are two uploading buffers I_1 and I_2 and two unloading buffers O_1 and O_2 in order to upload and download RAS by two raw product types $Pr1$ and $Pr2$. They are processed by machine M and moved by robot R_1 (resp. R_2) and R_2 (resp. R_1). There are two production cycles: (i) a raw product $Pr1$ is taken from I_1 by R_2 and put in M . After being processed by M , the product is unloaded by R_1 and put to O_1 .

(ii) a raw product $Pr2$ is taken from I_2 by R_1 and put in M . After being processed by M it is moved from M to O_2 by R_2 .

The PN model of this RAS is given in Figure 8. RT of the PN model is too large for displaying here, because it has 216 nodes (including \mathbf{x}_0). It (more precisely its adjacency matrix) points out on 5 deadlocks - \mathbf{x}_{54} , \mathbf{x}_{58} , \mathbf{x}_{62} , \mathbf{x}_{86} , \mathbf{x}_{121} .

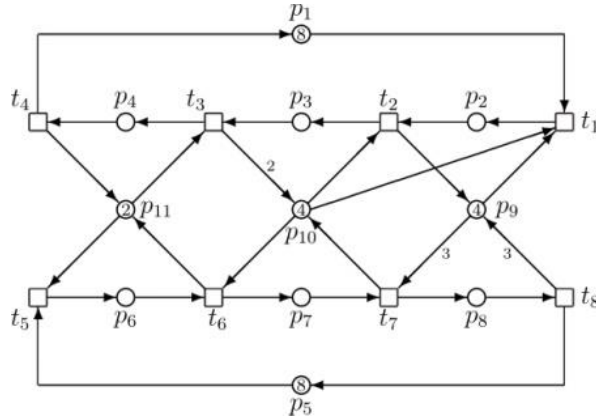


Figure 8
The PN model of RAS

4.3.1 P-invariant Based Approach

Let us apply the approach based on P -invariants to resolve the problem of deadlock avoidance.

$$\mathbf{x}_0 = (8 \ 0 \ 0 \ 0 \ 8 \ 0 \ 0 \ 0 \ 4 \ 4 \ 2)^T; \mathbf{x}_{54} = (4 \ 4 \ 0 \ 0 \ 6 \ 2 \ 0 \ 0 \ 0 \ 0 \ 0)^T$$

$$\mathbf{x}_{58} = (5 \ 2 \ 1 \ 0 \ 6 \ 2 \ 0 \ 0 \ 2 \ 0 \ 0)^T; \mathbf{x}_{62} = (6 \ 0 \ 2 \ 0 \ 6 \ 2 \ 0 \ 0 \ 4 \ 0 \ 0)^T$$

$$\mathbf{x}_{86} = (5 \ 3 \ 0 \ 0 \ 5 \ 2 \ 1 \ 0 \ 1 \ 0 \ 0)^T; \mathbf{x}_{121} = (6 \ 2 \ 0 \ 0 \ 4 \ 2 \ 2 \ 0 \ 2 \ 0 \ 0)^T$$

The structural matrix and \mathbf{x}_0 of the original uncontrolled system are the following:

$$\mathbf{B} = \begin{pmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 \\ -4 & 4 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & -4 & 4 & -2 & 2 & 0 & -1 & 1 & 1 \end{pmatrix}$$

$$\mathbf{x}_0 = (8 \ 0 \ 0 \ 0 \ 8 \ 0 \ 0 \ 0 \ 4 \ 4 \ 2)^T$$

After analysis between deadlocks and relative states (nodes RT) we can put \mathbf{L} and \mathbf{b} as follows:

$$\mathbf{L} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}; \mathbf{b} = \begin{pmatrix} 2 \\ 1 \\ 2 \\ 1 \\ 12 \end{pmatrix}$$

Hence, we obtain the structure and the initial state of the supervisor:

$$\mathbf{F}_s = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}; \mathbf{G}_s^T = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{x}_{s0} = (2 \ 1 \ 2 \ 1 \ 4)^T$$

The PN model of the supervised system is displayed in Figure 9. No deadlocks occur there in the supervised system.

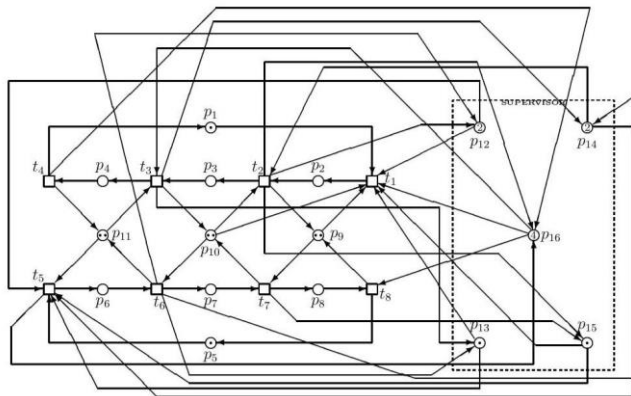


Figure 9

The PN model of the controlled RAS

5 Approach Based on Siphons and Traps

At this approach minimal siphons and minimal traps are computed. It may be realized e.g. using the tool [20] in Matlab. However, in general, at more complicated structure of the PN model and a big number of PN places, such approach may be also fairly time-consuming. This is valid for this tool too.

When we have minimal siphons in the matrix form \mathbf{S}_M (with particular siphons being its rows), we can obtain the supervisor structure as follows:

$$\mathbf{B}_s = \mathbf{S}_M \cdot \mathbf{B}; \quad \mathbf{F}_s = -\mathbf{B}_s^{(-)}; \quad \mathbf{G}_s^T = \mathbf{B}_s^{(+)} \quad (21)$$

5.1 Application on Example 1

First of all, let us apply this approach on Example 1 in the Subsection 3.1. Let us resolve the problem with the deadlock in it by this way. The minimal siphons are the rows of the following matrix:

$$\mathbf{S}_M = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Consequently, using the structural matrix \mathbf{B} of the original system we obtain the structural matrix of the supervisor:

$$\mathbf{B}_s = \mathbf{S}_M \cdot \mathbf{B} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -3 & 3 & -1 & 1 & 0 & 0 & 0 & 0 \\ -4 & 1 & 3 & -2 & 2 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{x}_{0s} = (0 \ 0 \ 3 \ 0 \ 3 \ 4)^T$$

The PN model of the controlled RAS is given in Figure 14.

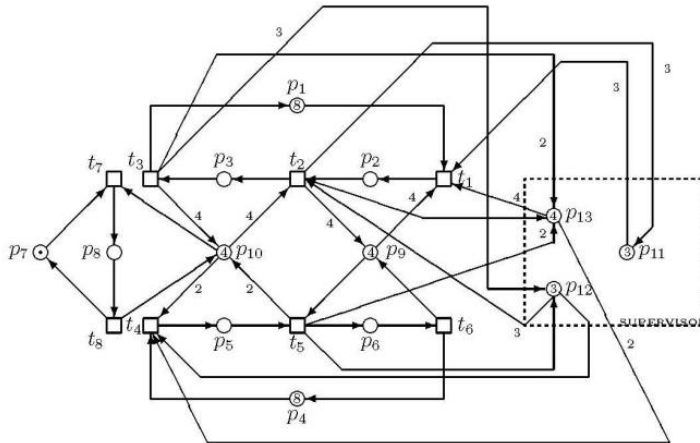


Figure 10
The PN model of the controlled RAS

5.2 Application on Example 2

Now, apply such an approach on the Example 2 analyzed in the Subsection 4.3. There are 199 siphons in the uncontrolled PN model and 8 minimal siphons (rows of the matrix \mathbf{S}_M) and 8 minimal traps (rows of \mathbf{T}_M), namely:

$$\mathbf{S}_M = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

$$\mathbf{T}_M = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

As we can see at comparing \mathbf{S}_M an \mathbf{T}_M , minimal siphons $S_1 = Tr_1$, $S_3 = Tr_2$, $S_4 = Tr_3$, $S_7 = Tr_6$. Because these traps are marked and they cannot lose tokens, the corresponding siphons cannot stay deadlocks. It means, that at synthesizing of the supervisor it is sufficient to use only siphons as follows:

$$\mathbf{S} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Then, the structural matrix of the supervisor \mathbf{S} is as follows:

$$\mathbf{B}_s = \mathbf{G}_s^T - \mathbf{F}_s = \mathbf{S} \cdot \mathbf{B} = \begin{pmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & -1 & 1 & 0 & 0 \\ -2 & 1 & 1 & 0 & 0 & -1 & -1 & 2 \\ -2 & 0 & 2 & 0 & -1 & 0 & -1 & 2 \end{pmatrix}$$

where $\mathbf{G}_s^T = \mathbf{B}_s^{(+)}$, while $\mathbf{F}_s = |\mathbf{B}_s^{(-)}|$.

For the initial state \mathbf{x}_0 of the uncontrolled PN model the initial state of the supervisor is $\mathbf{x}_{0s} = (4 \ 6 \ 8 \ 10)^T$. The PN model of the supervised system is displayed in Fig. 11. To verify if the supervised system is deadlock-free, we can compute RT starting from \mathbf{x}_0 . It has 95 nodes. No deadlocks were found in RT.

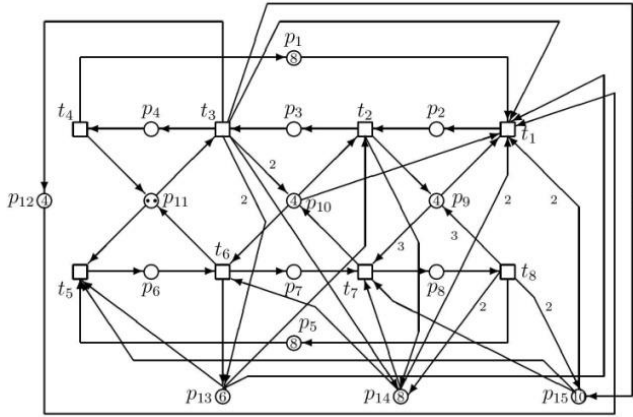


Figure 11

The PN model of the controlled RAS. The controller is created by the four places $p_{12} - p_{15}$

6 Comparison of the Presented Approaches

The approach in Section 3, using additional transitions, is only an auxiliary approach, but it has its importance, in cases with several diamonds in RT. Both principal approaches, presented in Section 4 and Section 5, respectively, are general and have their advantages and disadvantages. Moreover, the P -invariant based approach and the siphons and traps based one are much more powerful than that in Section 3. Therefore, only these two approaches will be compared here. The P -invariant based approach is more detailed because it works with PN places at the formulation of the matrix \mathbf{L} and the vector \mathbf{b} . Suitable combinations of places and assigning them maximal common numbers of tokens is very useful. On the other hand, at large or complicated PN models the computing of RT may be time-consuming and analyzing RT may be complicated. The siphons and traps based approach, does not need to compute RT, but only the siphons and traps have to be computed. However, the computation of them also depends on the size and especially on the complicity of the PN model structure. This process can also be time-consuming, even more than that in previous case. Both of the compared approaches are very useful at synthesizing supervisor for deadlocked DES and especially RAS. Thus, they are able to avoid deadlocks and simultaneously, successfully control RAS.

Nevertheless, it is necessary to take into account the computational complexity of algorithms in both approaches, especially in the case of greater number of places and complicated structure of the PN model. The complexity at computation RT is between the upper bound $2^{O(n \log n)}$ and the lower bound $2^{O(\sqrt{n})}$, where n is a number of PN places. At computing minimal siphons, the complexity is $O(2n + k(k - 1)/2)$,

where n and k denote, respectively, the number of PN places and the number of all PN siphons. However, all PN siphons can be found with the computational complexity $O(2^n)$. Of course, the computational time depends on the hardware ability of a computer in question, however, the computational complexity of algorithms is unchanged.

Conclusions

With regard to the aims declared in Subsection 1.2, three approaches for control deadlocked DES were introduced in this paper. One of them, using insertion of additional transitions, is in effect, auxiliary but sometimes useful. The further two approaches, are very useful for avoiding deadlocks in DES and simultaneously for controlling them. For two examples, all approaches were applied and illustrated. It can be said that both of the essential approaches are very appropriate. Comparisons for them were also introduced. To declare unambiguously, which is better, further research is needed, especially testing larger and more complicated structures of DES and thus, more knowledge can be obtained. From the introduced examples, is clear, that the siphon and traps based approach, is more practical, since it employs one-stage and does not need any further computations. On the other hand, the RT based approach, uses two-stages. It needs deep analysis of RT and then, to set corresponding conditions, for the matrix \mathbf{L} creation. A large RT cannot be seen in graphical form but only in the form of its adjacency matrix, which is also very large. Therefore, the creation of the matrix \mathbf{L} may be time consuming and even sometimes, impossible. Conversely, the computational complexity of the siphon based approaches, is better at calculating RT. In a large RAS, it can lead to the long computational times. Hence, in general, new approaches with smaller computational complexity have to be explored.

As to our plans for further research, the utilization of the mixed integer programming (MIP) will be tested in the siphon based approach, for the control of RAS, since the interesting area seems to be discovering robust algorithms for the RAS control.

Acknowledgement

This work was partially supported by the Slovak Grant Agency for Science VEGA under Grant No. 2/0020/21.

References

- [1] J. C. Luo, Z. Q. Liu, M. C. Zhou: A Petri Net Based Deadlock Avoidance Policy for Flexible Manufacturing Systems with Assembly Operations and Multiple Resource Acquisition. *IEEE Transactions on Industrial Informatics*, Vol. 15, No. 6, 2019, pp. 3379-3387
- [2] S. A. Reveliotis: Logical Control of Complex Resource Allocation Systems. *Foundations and Trends in Systems and Control*, Vol. 4, No. 1-2, 2017, pp. 1-223

-
- [3] M. A. Lawley, S. A. Reveliotis: Deadlock Avoidance for Sequential Resource Allocation Systems: Hard and Easy Cases. *The International Journal of Flexible Manufacturing Systems*, Vol. 13, 2001, pp. 385-404
- [4] S. A. Reveliotis: Coordinating Autonomy: Sequential Resource Allocation Systems for Automation. *IEEE Robotics & Automation Magazine*, Vol. 22, No. 2, 2015, pp. 77-94, DOI: 10.1109/MRA.2015.2401295
- [5] M. W. Iordache, P. J. Antsaklis: Supervision Based on Place Invariants: A Survey, *Discrete Event Dynamic Systems*, Vol. 16, 2006, pp. 451-492
- [6] Z. W. Li, M. C. Zhou: *Deadlock Resolution in Automated Manufacturing Systems. A Novel Petri Net Approach*. Springer-Verlag London Ltd., 2009
- [7] J. C. Luo, Z. Q. Liu, M. C. Zhou: A Petri Net Based Deadlock Avoidance Policy for Flexible Manufacturing Systems with Assembly Operations and Multiple Resource Acquisition. *IEEE Transactions on Industrial Informatics*, Vol. 15, No. 6, 2019, pp. 3379-3387
- [8] Z. Y. Ma, Z. W. Li, A. Giua: Petri net Controllers for Generalized Mutual Exclusion Constraints with Floor Operators. *Automatica*, Vol. 74, 2016, pp. 238-246
- [9] M. M. Yan, Z. W. Li, N. Wei, M. Zhao: A Deadlock Prevention Policy for a Class of Petri Nets S^3PMR . *Journal of Information Science and Engineering*, Vol. 25, 2009, pp. 167-183
- [10] G. Y. Liu, K. Barkaoui: A Survey of Siphons in Petri Nets. *Information Sciences*, Vol. 363, No. 1, 2016, pp. 198-220
- [11] F. Tricas, J. Ezpeleta: Computing Minimal Siphons in Petri Net Models of Resource Allocation Systems: A Parallel Solution, *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, Vol. 36, No. 3, May 2006, pp. 532-539, DOI: 10.1109/TSMCA.2005.855751
- [12] A. Giua, M. Silva: Modeling, Analysis and Control of Discrete Event Systems: A Petri Net Perspective. In: 20th IFAC World Congress, Toulouse, France, IFAC PapersOnLine 50-1, 2017, pp. 1772-1783
- [13] M. P. Fanti, B. Maione, S. Mascolo, B. Turchiano: Event-Bases Feedback Control for Deadlock Avoidance in Flexible Production Systems. *IEEE Transaction on Robotics and Automation*, Vol. 13, No. 3, 1997, pp. 347-363, DOI: 10.1109/70.585898
- [14] Z. Y. Ma, Z. Li, A. Giua: Petri net Controllers for Generalized Mutual Exclusion Constraints with Floor Operators. *Automatica*, Vol. 74, 2016, pp. 238-246
- [15] J. Desel, W. Reisig: Place/Transition Petri Nets. In: W. Reisig, G. Rozenberg (Eds.): *Advances of Petri Nets, Lecture Notes in Computer Science*, Vol. 1491, Springer, Heidelberg, 1998, pp. 122-173

-
- [16] G. Y. Liu, D. Y. Chao, F. Yu: Control Policy for a Subclass of Petri Nets without Reachability Analysis. IET Control Theory and Applications, Vol. 7, No. 8, 2013, pp. 1131-1141
- [17] T. C. Row, W. M. Syu, Y. L. Pan, C. C. Wang: One Novel and Optimal Deadlock Recovery Policy for Flexible Manufacturing Systems Using Iterative Control Transitions Strategy. Mathematical Problems in Engineering, Vol. 2019, Article ID 4847072, 12 p.
- [18] H. Yue, K. Y. Xing, H. S. Hu, W. M. Wu, H. Y. Su: PetriNet-Based Robust Supervisory Control of Automated Manufacturing Systems. Control Engineering Practice, Vol. 54, 2016, pp. 176-189
- [19] F. Čapkovič: Modelling and Control of Discrete-Event Systems with Partial Non-Determinism Using Petri Nets, Acta Polytechnica Hungarica, Vol. 17, No. 4, 2020, pp. 47-66
- [20] R. Davidrajuh: GPenSIM, General purpose Petri Net Simulator for Matlab Platform. Available: <http://www.davidrajuh.net/gpensim/>