Monitor design for Colored Petri Nets: an application to deadlock prevention in railway networks*

Maria Pia Fanti
Dip. di Elettrotecnica ed Elettronica
Politecnico di Bari, Italy
fanti@deemail.poliba.it

Alessandro Giua, Carla Seatzu
Dip. di Ing. Elettrica ed Elettronica
Università di Cagliari, Italy
{giua,seatzu}@diee.unica.it

Abstract

In this paper we use Colored Petri Nets (CPN) to model the dynamics of a railway system: places represent tracks and stations, tokens are trains. Using digraph tools, deadlock situations are characterized and a strategy is established to define off-line a set of constraints that prevent deadlocks. We show that these constraints limit the weighted sum of colored tokens in subsets of places. In particular, we extend the notion of Generalized Mutual Exclusion Constraints (GMEC) to CPN and we show that the above constraints, as well as the collision avoidance constraints, can be written as colored GMEC.

To solve this problem, we extend the theory of monitor places for place/transition nets to the case of CPN and we show that these constraints can be enforced by a colored monitor place that minimally restricts the behaviour of the closed-loop system.

Keywords: Colored Petri nets, generalized mutual exclusion constraints, monitor places, railway networks, deadlock prevention.

*Accepted for publication in *Control Engineering Practice.* Corresponding author is **Alessandro Giua**.

1 Introduction

Railways form one of the most important part of transportation systems and their constantly improving safety record, makes them a very attractive option compared with other modes of transport [16, 18, 24, 25, 27]. As a result, the overall complexity of railway network systems (RNS) increases, and hence greater demands are placed on the control logic of these systems [18].

Consider, to mention just one example, the case of the European Union. As the EU is opening to counties of Eastern Europe, it is also making substantial investments to revitalize the railways and plans to achieve the following objectives by 2020 [4]:

- for rail to increase its market share of passenger traffic from 6% to 10% and of goods traffic from 8% to 15%;
- a trebling of manpower productivity on the railways;
- a 50% gain in energy efficiency;
- a 50% reduction in emissions of pollutants;
- an increase in infrastructure capacity commensurate with traffic targets.

The specification, analysis and implementation of control logic for RNS is an important activity because its failure can lead to railway accidents and loss of human life [26]. At present time, this activity is even more important because railway networks are often large, the speed of trains and traffic density is increasing, and activities within networks are taking place concurrently and at geographically different locations [18, 24].

Very different approaches have been used to design efficient controllers for RNS. For a detailed treatment of the subject, the interested reader may consult the literature [16, 18, 24, 25, 27] and follow links provided on Internet sites [2, 3]. In [5] railway networks are modelled as discrete event systems that define a control design problem leading to a non-convex nonlinear optimization problem.

Note that the control of a railway network can be divided into two parts: *logical control* and *performance control*. The first one deals with structural problems, and imposes the satisfaction of a series of safeness constraints (collision avoidance) and liveness constraints (deadlock freeness). The second one, is related to the operation of the network and is concerned with problems such as scheduling both the departures and the stops, so as to optimize the efficiency of the net.

In this paper the attention is uniquely devoted to the design of control logic for logical control. The contribution of this paper is threefold.

1. We discuss how a RNS can be modelled with a particular class of Petri nets, called *Colored Petri nets* (CPN) [19, 20], that provide a powerful framework to the analysis and the definition of safeness constraints. In CPN, attributes (called colors) are associated to tokens, so that different activities can be assigned to tokens of different types, within the same structure of the net. The main advantage of CPN with respect to (wrt) other discrete event models, such as place/transition nets and Finite State Machines (FSM), is that this

model carries the relative simplicity and graphical representation of the other approaches, along with greater support for concurrency.

In [12] the railway network is modelled by a Petri net (PN) and deadlock avoidance constraints are expressed as *Generalized Mutual Exclusion Constraints* (GMEC). However, the controller does not distinguish among the routes assigned to the trains, and when forks and joins are present in the network the marking does not describe completely the system state.

This paper overcomes this problem by using CPN. More precisely, places represent tracks and stations, while transitions are the control points where the train movements are enabled or inhibited. The trains travelling in the system are represented by colored tokens and their color is the assigned path.

2. In this paper we consider two types of constraints, namely *collision prevention* constraints and *deadlock prevention* constraints.

   Collision prevention constraints originates from the fact that all tracks and stations have a finite capacity, thus we have to ensure that each resource does not accomodate a number of trains that is greater than the corresponding capacity.

   Deadlock prevention constraints ensure that no deadlock situation may occur. To characterize deadlock situations we use some results obtained in the field of Automated Manufacturing Systems where deadlock has been extensively studied [6, 7, 8, 17, 28]. Simple and efficient deadlock avoidance policies exist in the related literature (e.g., [1]).

   The aim of this paper is that of proposing a deadlock control function that has to be applied to particular systems in which, for the sake of security, the decision about the resource acquisition can not be taken in real time, by means of a deadlock avoidance policy. Hence, we characterize deadlock states by using digraph tools that describe the interactions between trains and resources (i.e., tracks and stations). In addition, a *deadlock prevention strategy* defines off-line the rules to prevent deadlock in advance.

   We observe that both collision and deadlock prevention constraints have a particular structure, namely, they limit the weighted sum of colored tokens in subsets of places. To this aim we extend the notion of GMEC to the case of CPN and we show that all these constraints can be written as colored GMEC.

3. The third contribution of this paper consists in showing that the control approach based on the construction of monitor places presented in [11] for place/transition nets can be extended to the more general case of CPN [19]. In particular, we show that when all transitions are controllable and observable, a colored GMEC can still be enforced by adding a colored monitor place p_c. We also provide a systematic procedure to compute the incidence matrix defining such a monitor place, as well as its initial marking. Moreover, we show that under the assumption that all transitions are controllable and observable, the monitor place minimally restricts the behavior of the closed-loop system, in the sense that it prevents only those transition firings that yield forbidden markings.

   Finally we apply these results to design a controller for the considered RNS.

Note that the supervisory design approach based on colored GMEC and monitor places can be easily extended to the case of transitions that are uncontrollable and/or unobservable wrt

certain colors, i.e., it can be easily extended to the case in which the monitor designed for a given GMEC is not admissible (it either disables a transition wrt an uncontrollable color, or observes the firing of a transition wrt an unobservable color). In particular, in [12] we showed that in such a case one can design a less permissive, but admissible monitor, extending to the case of colored nets the parametrization and the tabular procedure proposed by Moody and Antsaklis for PN [22]. This point is not dealt here for sake of brevity. The interested reader is referred to [12] for a comprehensive discussion of this approach.

RNS are just one of the many application fields in which it is much more convenient to deal with a CPN model rather than a PN model. In all these cases an alternative to our procedure would be that of unfolding the net, converting the colored GMEC into a set of uncolored GMEC, then compute the admissible monitor places, and finally convert them into a single colored monitor place. Our approach provides a systematic procedure to compute the final colored monitor within the framework of CPN. Furthermore there is some computational advantage in our procedure: to compute a monitor for a colored GMEC, we do not need to compute the unfolding of the whole CPN, but we only consider the incidence matrix of the subnet composed by the places in the support of the GMEC.

Finally, the proposed approach is a first step towards the formulation of a procedure for the definition of GMEC and monitor places for arbitrary high-level PN.

The paper is structured as follows. In Section 2 we provide the formal definition of CPN based on the notion of multisets and on the matrix representation of multisets presented in the Appendix. The notion of GMEC is extended to the case of CPN in Section 3. The theory of monitors for CPN is proposed in Section 4. In Section 5 we define the considered RNS and we show how it can be easily modeled as a CPN. In this section we also introduce the collision constraints and we show how they can be imposed using monitors. The deadlock prevention policy is then derived in Section 6, based on the theory of digraphs. Conclusions are finally drawn in Section 7.

The paper contains 3 appendices. Appendix A recall the notion of multiset. Appendix B is a list of abbreviations. Appendix C is a list of symbols used in the paper.

Note that the main results of this paper have also been presented in two conference papers [9, 10].

2 Colored Petri nets

A *Colored Petri Net* (CPN) is a bipartite directed graph represented by a quintuple $N = (P, T, Co, \boldsymbol{Pre}, \boldsymbol{Post})$ where P is the set of places, T is the set of transitions, $Co : P \cup T \to \mathcal{C}l$ is a color function that associates to each element in $P \cup T$ a non empty ordered set of colors in the set of possible colors $\mathcal{C}l$.

Therefore, for all $p_i \in P$, $Co(p_i) = \{a_{i,1}, a_{i,2}, \ldots, a_{i,u_i}\} \subseteq \mathcal{C}l$ is the ordered set of possible colors of tokens in p_i, and u_i is the number of possible colors of tokens in p_i. Analogously, for all $t_j \in T$, $Co(t_j) = \{b_{j,1}, b_{j,2}, \ldots, b_{j,v_j}\} \subseteq \mathcal{C}l$ is the ordered set of possible occurrence colors of t_j, and v_j is the number of possible occurrence colors in t_j.

In the following we assume that $m = |P|$ and $n = |T|$.

Matrices $\boldsymbol{Pre}$ and $\boldsymbol{Post}$ are the pre-incidence and the post-incidence $m \times n$ dimensional matrices

respectively. In particular, each element $\boldsymbol{Pre}(p_i, t_j)$ is a mapping from the set of occurrence colors of t_j to a non negative multiset[1] over the set of colors of p_i, namely, $\boldsymbol{Pre}(p_i, t_j) : Co(t_j) \rightarrow \mathcal{N}(Co(p_i))$, for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. In the following we denote $\boldsymbol{Pre}(p_i, t_j)$ as a matrix of $u_i \times v_j$ non negative integers, whose generic element $Pre(p_i, t_j)(h, k)$ is equal to the weight of the arc from place p_i wrt color $a_{i,h}$ to transition t_j wrt color $b_{j,k}$.

Analogously, $\boldsymbol{Post}(p_i, t_j) : Co(t_j) \rightarrow \mathcal{N}(Co(p_i))$, for $i = 1, \ldots, m$ and $j = 1, \ldots, n$, and we denote $\boldsymbol{Post}(p_i, t_j)$ as a matrix of $u_i \times v_j$ non negative integers. The generic element $Post(p_i, t_j)(h, k)$ is equal to the weight of the arc from transition t_j wrt color $b_{j,k}$ to place p_i wrt color $a_{i,h}$.

The incidence matrix $\boldsymbol{C}$ is an $m \times n$ matrix, whose generic element $\boldsymbol{C}(p_i, t_j) : Co(t_j) \rightarrow \mathcal{Z}(Co(p_i))$, for $i = 1, \ldots, m$ and $j = 1, \ldots n$. In particular $\boldsymbol{C}(p_i, t_j) = \boldsymbol{Post}(p_i, t_j) - \boldsymbol{Pre}(p_i, t_j)$.

For each place $p_i \in P$, we define the *marking* $\boldsymbol{m}_i$ of p_i as a *non negative multiset* over $Co(p_i)$. The mapping $m_i : Co(p_i) \rightarrow \mathbb{N}$ associates to each possible token color in p_i a non negative integer representing the number of tokens of that color that is contained in place p_i, and

$$\boldsymbol{m}_i = \sum_{d \in Co(p_i)} m_i(d) \otimes d.$$

In the following we denote $\boldsymbol{m}_i$ as a column vector of u_i non negative integers, whose h-th component $m_i(h)$ is equal to the number of tokens of color $a_{i,h}$ that are contained in p_i.

Finally, the marking $\boldsymbol{M}$ of a CPN is an m-dimensional column vector of multisets, i.e.,

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{m}_1 \\ \vdots \\ \boldsymbol{m}_m \end{bmatrix}.$$

A CPN system $\langle N, \boldsymbol{M}_0 \rangle$ is a CPN N with initial marking $\boldsymbol{M}_0$.

A transition $t_j \in T$ is *enabled* wrt color $b_{j,k}$ at a marking $\boldsymbol{M}$ if and only if for each place $p_i \in P$ and for all $h = 1, \ldots, u_i$, we have $m_i(h) \geq Pre(p_i, t_j)(h, k)$.

If an enabled transition t_j fires at $\boldsymbol{M}$ wrt color $b_{j,k}$, then we get a new marking $\boldsymbol{M'}$ where, for all $p_i \in P$ and for all $h = 1, \ldots, u_i$, $m'_i(h) = m_i(h) + Post(p_i, t_j)(h, k) - Pre(p_i, t_j)(h, k)$.

We will write $\boldsymbol{M}[t_j(k)\rangle \boldsymbol{M'}$ to denote that t fires at $\boldsymbol{M}$ wrt color $b_{j,k}$ yielding $\boldsymbol{M'}$.

A *firing sequence* from $\boldsymbol{M}_0$ is a (possibly empty) sequence of transitions, each one firing wrt a given color,

$$\sigma = t_{j_1}(k_{j_1}) t_{j_2}(k_{j_2}) \ldots t_{j_r}(k_{j_r})$$

such that

$$\boldsymbol{M}_0[t_{j_1}(k_{j_1})\rangle \boldsymbol{M}_1[t_{j_2}(k_{j_2})\rangle \boldsymbol{M}_2 \ldots t_{j_r}(k_{j_r})\rangle \boldsymbol{M}_r.$$

A marking $\boldsymbol{M}$ is *reachable* in $\langle N, \boldsymbol{M}_0 \rangle$ iff there exists a firing sequence σ such that $\boldsymbol{M}_0[\sigma\rangle \boldsymbol{M}$.

Given a system $\langle N, \boldsymbol{M}_0 \rangle$, the set of firing sequences (also called *language* of the net) is denoted $L(N, \boldsymbol{M}_0)$ and the set of reachable markings (also called the *reachability set* of the colored net) is denoted $R(N, \boldsymbol{M}_0)$.

[1] In Appendix A we recall all the definitions and properties concerning multisets that are useful in the paper.

If the marking $\boldsymbol{M}$ is reachable in $\langle N, \boldsymbol{M}_0 \rangle$ by firing a sequence σ, then the following *state equation* is satisfied:

$$\boldsymbol{M} = \boldsymbol{M}_0 + \boldsymbol{C} \circ \boldsymbol{\Sigma}$$

where

$$\boldsymbol{\Sigma} = \left[\begin{array}{ccc} \boldsymbol{\sigma}_1 & \ldots & \boldsymbol{\sigma}_n \end{array} \right]^T$$

is a vector of non negative multisets, and $\boldsymbol{\sigma}_j \in \mathcal{N}(Co(t_j))$, for $j = 1, \ldots, n$ is a multiset that specifies how many times transition t_j has fired wrt each of its colors. The vector $\boldsymbol{\Sigma}$ is called the *firing count vector* of the firing sequence σ.

Finally, let $\boldsymbol{X}$ be an m-dimensional vector of multisets where for all $i = 1, \ldots, m$, $\boldsymbol{x}_i \in \mathcal{N}(Co(p_i))$. Let $P' \subseteq P$. The projection of $\boldsymbol{X}$ on P' is the restriction of $\boldsymbol{X}$ to P' and will be denoted $\boldsymbol{X} \uparrow_{P'}$. This definition is extended in the usual way to the projection of a set of vectors $\mathcal{X}$, i.e., $\mathcal{X} \uparrow_{P'} = \{\boldsymbol{X} \uparrow_{P'} \mid \boldsymbol{X} \in \mathcal{X}\}$.

Example 2.1. Let us consider the CPN in Figure 1.a apart from place p_c and all connected arcs. The set of colors is $\mathcal{C}l = \{c_1, c_2, c_3\}$. Place p_1 may only contain tokens of colors c_2 and c_3, while place p_2 may contain tokens of any color in $\mathcal{C}l$. Finally, transitions t_1 and t_3 may only fire wrt to colors c_1 and c_2, while transition t_2 may fire wrt any color in $\mathcal{C}l$.

Given the structure of the net, the only non null matrices $\boldsymbol{Pre}$ and $\boldsymbol{Post}$ are those reported Figure 1.a using the matrix notation, or equivalently,

$$\begin{array}{l}
\boldsymbol{Post}(p_1, t_1)(c_1) = 2 \otimes c_2 + 1 \otimes c_3, \\
\boldsymbol{Post}(p_1, t_1)(c_2) = 1 \otimes c_2 + 2 \otimes c_3, \\
\boldsymbol{Pre}(p_1, t_2)(c_1) = 1 \otimes c_2 + 1 \otimes c_3, \\
\boldsymbol{Pre}(p_1, t_2)(c_2) = 2 \otimes c_2 + 1 \otimes c_3, \\
\boldsymbol{Pre}(p_1, t_2)(c_3) = 1 \otimes c_2 + 2 \otimes c_3, \\
\boldsymbol{Post}(p_2, t_2)(c_1) = 1 \otimes c_1 + 1 \otimes c_3, \\
\boldsymbol{Post}(p_2, t_2)(c_2) = 2 \otimes c_2, \\
\boldsymbol{Post}(p_2, t_2)(c_3) = 1 \otimes c_1 + 1 \otimes c_2 + 1 \otimes c_3, \\
\boldsymbol{Pre}(p_2, t_3)(c_1) = 1 \otimes c_1 + 2 \otimes c_2, \\
\boldsymbol{Pre}(p_2, t_3)(c_2) = 2 \otimes c_1 + 1 \otimes c_2 + 1 \otimes c_3.
\end{array}$$

Assuming that no token is initially contained in the net, i.e.,

$$\boldsymbol{M}_0 = \left[\begin{array}{cc} \boldsymbol{\varepsilon} & \boldsymbol{\varepsilon} \end{array} \right]^T,$$

if t_1 fires wrt to c_1 then we reach a new marking

$$\boldsymbol{M}_1 = \left[\begin{array}{cc} 2 \otimes c_2 + 1 \otimes c_3 & \boldsymbol{\varepsilon} \end{array} \right]^T.$$

Now, if t_2 fires wrt c_2, then we reach a new marking

$$\boldsymbol{M}_2 = \left[\begin{array}{cc} \boldsymbol{\varepsilon} & 2 \otimes c_2 \end{array} \right]^T.$$

The firing vector associated to the whole sequence $\sigma = t_1(c_1)t_2(c_2)$ is

$$\boldsymbol{\Sigma} = \left[\begin{array}{ccc} 1 \otimes c_1 & 1 \otimes c_2 & \boldsymbol{\varepsilon} \end{array} \right]^T.$$

■

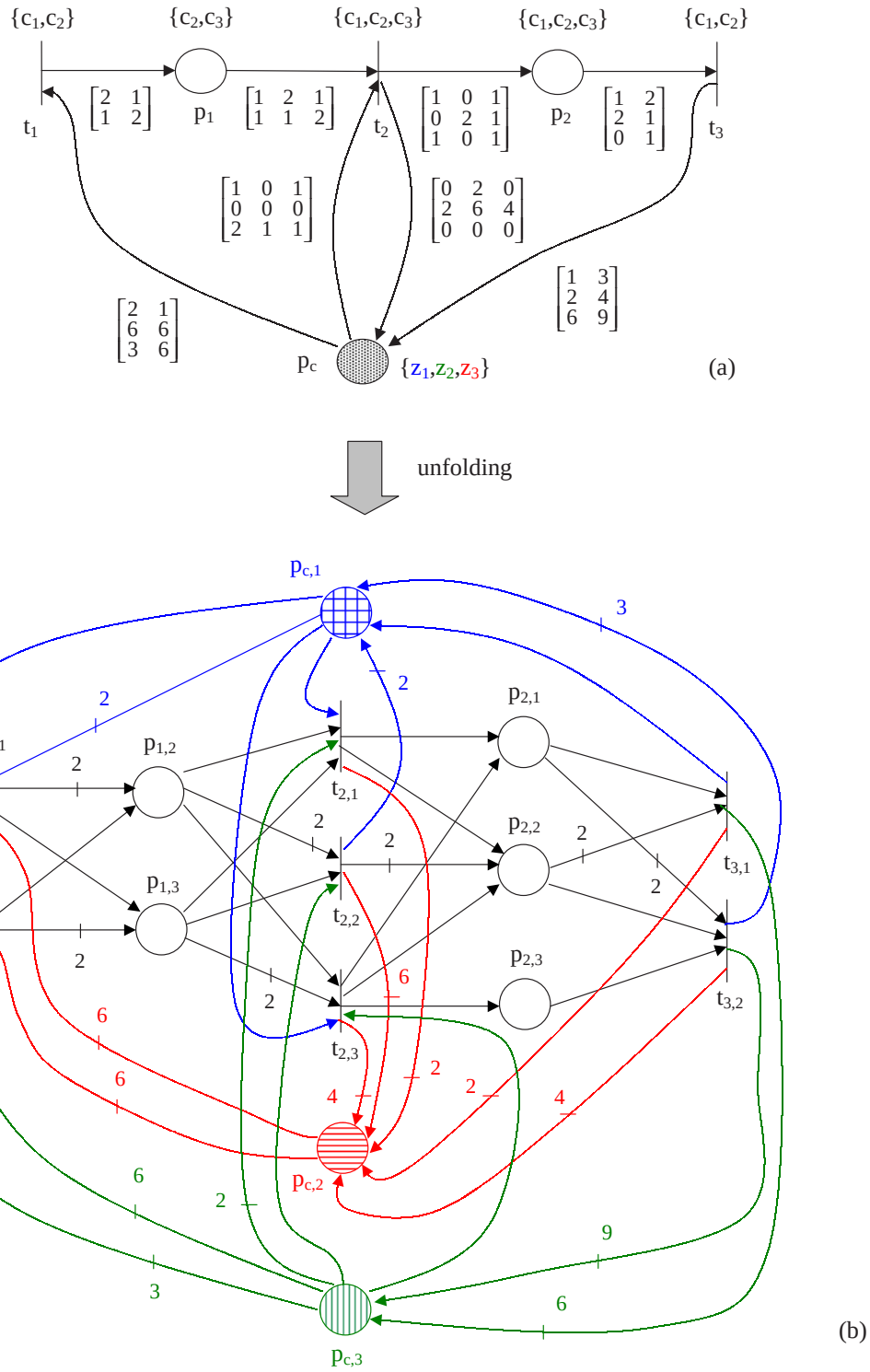

Figure 1: The closed-loop colored Petri net of Example 1 and the unfolded closed-loop net.

3 GMEC in colored Petri nets

In this section we extend the notion of GMEC for place/transition nets to the case of CPN. In fact, as it will be clear in the next session both collision and deadlock avoidance constraints can be written in the form of a colored GMEC. Then, in Section 4 we will show that when all transitions are controllable and observable, a GMEC can still be enforced by adding a monitor place p_c, and we provide a systematic procedure to compute the incidence matrix defining such a monitor place, as well as its initial marking.

3.1 GMEC in P/T nets

In the context of place/transition nets, the problem of designing a supervisory controller that restricts the open-loop reachability set $R(N_p, \boldsymbol{M}_{p,0})$ of a plant $\langle N_p, \boldsymbol{M}_{p,0}\rangle$, to a closed-loop reachability set $\mathcal{L} \cap R(N_p, \boldsymbol{M}_{p,0})$, where $\mathcal{L} \subseteq \mathbb{N}^m$ is a given set of *legal markings*, has been extensively investigated by many authors [15, 23, 30, 31]. Of particular interest in many applications are those control problems where the set of legal markings $\mathcal{L}$ is expressed by a set of n_c linear inequality constraints called *Generalized Mutual Exclusion Constraints* [11]. Each GMEC is a couple $(\boldsymbol{w}, k)$ where $\boldsymbol{w} : P \to \mathbb{Z}$ is a $m \times 1$ weight vector and $k \in \mathbb{Z}$, and defines a set of legal markings: $\mathcal{L} = \mathcal{M}(\boldsymbol{w}, k) = \{\boldsymbol{M} \in \mathbb{N}^m \mid \boldsymbol{w}^T \boldsymbol{M} \leq k\}$.

A controlling agent, called supervisor, must ensure that only legal markings are reached. If all transitions are controllable and observable, the maximally permissive supervisor for a GMEC takes the form of a single *monitor place* p_c. If $\boldsymbol{C}_p$ is the incidence matrix of the open-loop plant and $\boldsymbol{M}_{p,0}$ is its initial marking, the monitor that enforces $(\boldsymbol{w}, k)$ has incidence matrix $\boldsymbol{C}_c = -\boldsymbol{w}^T \boldsymbol{C}_p$ and initial marking $m_{c,0} = k - \boldsymbol{w}^T \boldsymbol{M}_{p,0}$. The controller exists iff the initial marking $\boldsymbol{M}_{p,0}$ is a legal marking, i.e., $k - \boldsymbol{w}^T \boldsymbol{M}_{p,0} \geq \boldsymbol{0}$. By definition a monitor is loop-free[2], thus its incidence matrix $\boldsymbol{C}_c$ uniquely defines the pre- and post- incidence matrices $\boldsymbol{Pre}_c = \max\{-\boldsymbol{C}_c, \boldsymbol{0}\}$ and $\boldsymbol{Post}_c = \max\{\boldsymbol{C}_c, \boldsymbol{0}\}$.

3.2 GMEC in CPN

Now, we introduce the notion of *colored* GMEC: we show that it may represent in a compact way several constraints, and can be unfolded into a set of uncolored GMEC.

Definition 3.1. A GMEC is a couple $(\boldsymbol{W}, \boldsymbol{k})$ where

$$\boldsymbol{W} = \left[\begin{array}{ccc} \boldsymbol{w}_1 & \cdots & \boldsymbol{w}_m \end{array} \right], \qquad \boldsymbol{k} \in \mathcal{Z}(D), \tag{1}$$

for all i, $\boldsymbol{w}_i : Co(p_i) \to \mathcal{Z}(D)$, and D is a set of colors different from $Co(p_i)$, $i = 1, \cdots, m$. Thus $\boldsymbol{W}$ can also be represented by a matrix with $|D|$ rows and $\sum_{i=1}^{m} |Co(p_i)|$ columns.

[2] A transition cannot be at the same time input and output transition of a monitor.

The set of legal markings defined by $(\boldsymbol{W}, \boldsymbol{k})$ can be written as

$$\mathcal{M}(\boldsymbol{W}, \boldsymbol{k}) = \left\{ \boldsymbol{M} = \begin{bmatrix} \boldsymbol{m}_1 \\ \vdots \\ \boldsymbol{m}_m \end{bmatrix} \middle| \; \boldsymbol{m}_i \in \mathcal{N}(Co(p_i)), \right. \\ \left. \boldsymbol{W} \circ \boldsymbol{M} \triangleq \sum_{i=1}^{m} \boldsymbol{w}_i \circ \boldsymbol{m}_i \leq \boldsymbol{k} \right\}. \tag{2}$$

■

Note that here we are extending the $\circ$ operator to the case of scalar product of vectors of multisets.

Example 3.2. Let us consider again the CPN in Figure 1.a apart from place p_c and all connected arcs. Assume $D = \{z_1, z_2, z_3\}$. Moreover, let

$$\begin{aligned} &\boldsymbol{w}_1 = \begin{bmatrix} \boldsymbol{w}_1(c_2) & \boldsymbol{w}_1(c_3) \end{bmatrix} \\ &\boldsymbol{w}_1(c_1) = 1 \otimes z_1 + 2 \otimes z_2, \\ &\boldsymbol{w}_1(c_3) = 2 \otimes z_2 + 3 \otimes z_3, \end{aligned}$$

$$\begin{aligned} &\boldsymbol{w}_2 = \begin{bmatrix} \boldsymbol{w}_2(c_1) & \boldsymbol{w}_2(c_2) & \boldsymbol{w}_2(c_3) \end{bmatrix} \\ &\boldsymbol{w}_2(c_1) = 1 \otimes z_1 + 2 \otimes z_2 + 2 \otimes z_3, \\ &\boldsymbol{w}_2(c_2) = 2 \otimes z_3, \\ &\boldsymbol{w}_2(c_3) = 1 \otimes z_1 + 3 \otimes z_3. \end{aligned}$$

$$\boldsymbol{k} = 3 \otimes z_1 + 5 \otimes z_2 + 6 \otimes z_3.$$

Using the matrix notation we can write:

$$\boldsymbol{w}_1 = \begin{matrix} c_2 & c_3 \end{matrix} \\ \begin{bmatrix} 1 & 0 \\ 2 & 2 \\ 0 & 3 \end{bmatrix} \begin{matrix} z_1 \\ z_2 \\ z_3 \end{matrix} \qquad \boldsymbol{w}_2 = \begin{matrix} c_1 & c_2 & c_3 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 1 \\ 2 & 0 & 0 \\ 2 & 2 & 3 \end{bmatrix} \begin{matrix} z_1 \\ z_2 \\ z_3 \end{matrix}$$

and

$$\boldsymbol{k} = \begin{bmatrix} 3 & 5 & 6 \end{bmatrix}^T.$$

Therefore,

$$\boldsymbol{W} \circ \boldsymbol{M} \triangleq \sum_{i=1}^{m} \boldsymbol{w}_i \circ \boldsymbol{m}_i = \begin{bmatrix} 1 & 0 \\ 2 & 2 \\ 0 & 3 \end{bmatrix} \cdot \begin{bmatrix} m_1(c_2) \\ m_1(c_3) \end{bmatrix} + \begin{bmatrix} 1 & 0 & 1 \\ 2 & 0 & 0 \\ 2 & 2 & 3 \end{bmatrix} \cdot \begin{bmatrix} m_2(c_1) \\ m_2(c_2) \\ m_2(c_3) \end{bmatrix} \leq \begin{bmatrix} 3 \\ 5 \\ 6 \end{bmatrix}$$

and

$$\begin{aligned} \mathcal{M}(\boldsymbol{W}, \boldsymbol{k}) = \{ \quad & \boldsymbol{M} = \begin{bmatrix} \boldsymbol{m}_1 \\ \boldsymbol{m}_2 \end{bmatrix} \; | \; \boldsymbol{m}_i \in \mathcal{N}(Co(p_i)), \\ & m_1(c_2) + m_2(c_1) + m_2(c_3) \leq 3, \\ & 2m_1(c_2) + 2m_1(c_3) + 2m_2(c_1) \leq 5 \\ & 3m_1(c_3) + 2m_2(c_1) + 2m_2(c_2) + 3m_2(c_3) \leq 6 \}. \end{aligned}$$

In the next section we show how to compute a monitor place p_c that can be added to the net in order to enforce the given specification. ■

4 Monitors for colored Petri nets

In this section we show how the results presented in [11] for place/transition nets can be extended to the more general case of CPN. In particular, we show that a GMEC can still be enforced by adding a monitor place p_c, and we provide a systematic procedure to compute the incidence matrix defining such a monitor place, as well as its initial marking.

Note that in this section we assume that all transitions are controllable and observable. This implies that the firing of all transitions with respect to any color may be prevented and observed by any external agent.

Definition 4.1. Given a CPN system $\langle N_p, \boldsymbol{M}_{p,0}\rangle$, with $N_p = (P, T, Co, \boldsymbol{Pre}_p, \boldsymbol{Post}_p)$, and a GMEC $(\boldsymbol{W}, \boldsymbol{k})$, the *monitor* that enforces this constraint is a new place p_c with $Co(p_c) = D$, to be added to N_p. The resulting system is denoted $\langle N, \boldsymbol{M}_0\rangle$, with $N = (P \cup \{p_c\}, T, Co, \boldsymbol{Pre}, \boldsymbol{Post})$. Then N will have incidence matrix

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{C}_p \\ \boldsymbol{C}_c \end{bmatrix}, \quad \text{where } \boldsymbol{C}_c = -\boldsymbol{W} \circ \boldsymbol{C}_p. \tag{3}$$

We are assuming that there are no self-loops containing p_c in N, hence $\boldsymbol{Pre}$ and $\boldsymbol{Post}$ may be uniquely determined by $\boldsymbol{C}$. This means that if

$$\boldsymbol{C}(p_i, t_j) = \sum_{k=1}^{v_j} c_k \otimes d_k$$

then

$$\boldsymbol{Post}(p_i, t_j) = \sum_{k=1}^{v_j} \max\{c_k, 0\} \otimes d_k, \quad \boldsymbol{Pre}(p_i, t_j) = \sum_{k=1}^{v_j} \max\{-c_k, 0\} \otimes d_k.$$

The initial marking of $\langle N, \boldsymbol{M}_0\rangle$ is

$$\boldsymbol{M}_0 = \begin{bmatrix} \boldsymbol{M}_{p,0} \\ \boldsymbol{m}_{c,0} \end{bmatrix}, \quad \text{where } \boldsymbol{m}_{c,0} = \boldsymbol{k} - \boldsymbol{W} \circ \boldsymbol{M}_{p,0}. \tag{4}$$

We assume that the initial marking $\boldsymbol{M}_{p,0}$ of the system satisfies the constraint $(\boldsymbol{W}, \boldsymbol{k})$. ■

In the case of controllable and observable transitions we can prove the following result.

Theorem 4.2 ([10])**.** Let $\langle N_p, \boldsymbol{M}_{p,0}\rangle$ be a CPN system, and $(\boldsymbol{W}, \boldsymbol{k})$ a colored GMEC. Let $\langle N, \boldsymbol{M}_0\rangle$ be the system with the addition of the monitor place p_c.

(1) The monitor place p_c enforces the GMEC $(\boldsymbol{W}, \boldsymbol{k})$ when included in the closed-loop system $\langle N, \boldsymbol{M}_0\rangle$.

(2) The monitor place p_c minimally restricts the behavior of the closed-loop system $\langle N, \boldsymbol{M}_0\rangle$, in the sense that it prevents only transition firings that yield forbidden markings.

Example 4.3. Let us consider again the CPN in Figure 1.a apart from place p_c and all connected arcs. The incidence matrix is

$$\boldsymbol{C}_p = \begin{bmatrix} \boldsymbol{Post}(p_1,t_1) & -\boldsymbol{Pre}(p_1,t_2) & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Post}(p_2,t_2) & -\boldsymbol{Pre}(p_2,t_3) \end{bmatrix}$$

where $\boldsymbol{Post}(p_1,t_1)$, $\boldsymbol{Pre}(p_1,t_2)$, $\boldsymbol{Post}(p_2,t_2)$ and $\boldsymbol{Pre}(p_2,t_3)$ are shown in Figure 1.a using the matrix notation introduced in Appendix.

Assume that we want to enforce the GMEC $(\boldsymbol{W},\boldsymbol{k})$ considered in the previous Example 3.2.

This constraint can be enforced by adding a monitor place p_c whose incidence matrix $\boldsymbol{C}_c$ is

$$\begin{aligned} \boldsymbol{C}_c &= \begin{bmatrix} \boldsymbol{C}_c(p_c,t_1) & \boldsymbol{C}_c(p_c,t_2) & \boldsymbol{C}_c(p_c,t_3) \end{bmatrix} \\ &= -\boldsymbol{W} \circ \boldsymbol{C}_p \\ &= -\begin{bmatrix} \boldsymbol{w}_1 & \boldsymbol{w}_2 \end{bmatrix} \circ \\ &\quad \begin{bmatrix} \boldsymbol{Post}(p_1,t_1) & -\boldsymbol{Pre}(p_1,t_2) & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Post}(p_2,t_2) & -\boldsymbol{Pre}(p_2,t_3) \end{bmatrix} \\ &= \begin{bmatrix} -(\boldsymbol{w}_1 \circ \boldsymbol{Post}(p_1,t_1))^T \\ (\boldsymbol{w}_1 \circ \boldsymbol{Pre}(p_1,t_2) - \boldsymbol{w}_2 \circ \boldsymbol{Post}(p_2,t_2))^T \\ (\boldsymbol{w}_2 \circ \boldsymbol{Pre}(p_2,t_3))^T \end{bmatrix}^T \\ &= \begin{array}{c} \begin{array}{cccccc} c_1 & c_2 & c_1 & c_2 & c_3 & c_1 \; c_2 \end{array} \\ \left[\begin{array}{cc|ccc|cc} -2 & 11 & -1 & 2 & -1 & 1 & 3 \\ -6 & -6 & 2 & 6 & 4 & 2 & 4 \\ -3 & -6 & -2 & -1 & -1 & 6 & 9 \end{array}\right] \end{array} \begin{array}{c} z_1 \\ z_2 \\ z_3 \end{array} \end{aligned}$$

The resulting closed-loop net is reported in Figure 1.a. For completeness in the same figure we have also reported the unfolding of the closed-loop net $\overline{N}$. Note that an uncolored Petri net is a CPN where for all $p \in P$ and for all $t \in T$, $Co(p) = Co(t) = \{\bullet\}$ where $\bullet$ is the usual uncolored token.

In particular, we used the following notation. Different colors and filling patterns have been used to denote the open-loop net and the monitor places. More precisely, black has been used to represent the open-loop net, while (squared) blue, r(horizontally striped) red and (vertically striped) green denote the monitor places ($p_{c,1}$, $p_{c,2}$, and $p_{c,3}$), and all arcs connected to them, relative to the constraints associated to colors z_1, z_2 and z_3, respectively. Moreover, the marking of the generic place $p_{i,j}$ denotes the number of tokens of color c_j that are contained in the place p_i of the original CPN. Finally, the firing of transition $t_{i,j}$ corresponds to the firing of transition t_i wrt color c_j. If we order the set of places and transitions of the unfolded open-loop net so that its marking $\overline{\boldsymbol{M}}$ is equal to

$$\overline{\boldsymbol{M}} = \begin{bmatrix} m_{1,2} & m_{1,3} & m_{2,1} & m_{2,2} & m_{2,3} \end{bmatrix}^T$$

and the set of transitions is

$$\overline{T} = \{t_{1,1}, t_{1,2}, t_{2,1}, t_{2,2}, t_{2,3}, t_{3,1}, t_{3,2}\},$$

the incidence matrix of the unfolded open-loop net is equal to

$$\overline{\boldsymbol{C}} = \begin{bmatrix} 2 & 1 & -1 & -2 & -1 & 0 & 0 \\ 1 & 2 & -1 & -1 & -2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & -1 & -2 \\ 0 & 0 & 0 & 2 & 1 & -2 & -1 \\ 0 & 0 & 1 & 0 & 1 & 0 & -1 \end{bmatrix}.$$

By looking at the definition of the set of consistent markings, it is easy to write the constraint matrix $\overline{\boldsymbol{W}}$, i.e.,

$$\overline{\boldsymbol{W}} = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 2 & 3 \\ 1 & 2 & 2 \\ 0 & 0 & 2 \\ 1 & 0 & 3 \end{bmatrix}.$$

Therefore the incidence matrix of the controller of the unfolded net is

$$\overline{\boldsymbol{C}}_c = -\overline{\boldsymbol{W}}^T \cdot \overline{\boldsymbol{C}} = \begin{bmatrix} -2 & -1 & -1 & 2 & -1 & 1 & 3 \\ -6 & -6 & 2 & 6 & 4 & 2 & 4 \\ -3 & -6 & -2 & -1 & -1 & 6 & 9 \end{bmatrix}$$

in accordance with the results obtained using the colored Petri net.

Now, assume that the initial marking of the open-loop colored net is

$$\boldsymbol{M}_{p,0} = \begin{bmatrix} 1 \otimes c_1 \\ 1 \otimes c_1 + 1 \otimes c_3 \end{bmatrix}$$

that satisfies the GMEC. In such a case the initial marking of the monitor place should be taken equal to

$$\begin{aligned} \boldsymbol{m}_{c,0} = \quad & \boldsymbol{k} - \boldsymbol{W} \circ \boldsymbol{M}_{p,0} = \boldsymbol{k} - \sum_{i=1}^{3} \boldsymbol{w}_i \circ \boldsymbol{m}_{p,0,i} \\ & = 1 \otimes z_1 + 3 \otimes z_2 + 1 \otimes z_3. \end{aligned}$$

Analogously, if we consider the unfolded net, using the well known theory of the GMECs, we find out that the initial marking of the monitor places is $m_{c,1} = m_{c,3} = 1$ and $m_{c,2} = 3$. ■

5 The railway network system

A RNS consists of three fundamental elements: railway lines, stations, vehicles (i.e., trains, single engines, etc.) travelling over these lines.

Let us consider the set $V = \{v_1, \cdots, v_{N_V}\}$ that collects all the vehicles moving over the lines and the stations. The railway lines are divided into several tracks and each track can be occupied by only one vehicle at a time. In our framework, each station is described by a resource r_i, for $i = 1, \cdots, N_S$, where N_S is the number of stations. Moreover, each track of the RNS is viewed as a resource that vehicles can acquire and it is denoted by r_i, for $i = N_S + 1, \cdots, N_S + N_T$, where N_T is the overall number of tracks. Since each station is composed of one or more tracks,

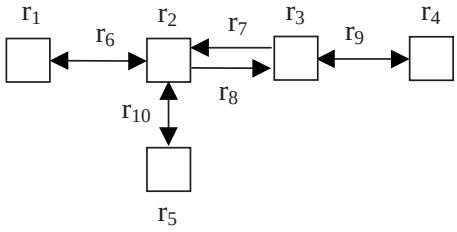

Figure 2: The railway network of Example 5.1.

and each track can accommodate only one train at a time, a finite capacity $C(r_i) \geq 1$ is assigned to each station r_i, for $i = 1, \cdots, N_S$. Moreover, each track r_i has unit capacity, i.e., $C(r_i) = 1$ for all $i = N_S + 1, \cdots, N_S + N_T$.

We also assume that the terminal stations of the train paths are connected to a "virtual" *docking station* r_0. The docking station can accommodate all the trains in the system, i.e., $C(r_0) = \infty$.

Finally, we generically call *resources* or *nodes* the stations and the tracks. Therefore, the set $R = \{r_i,\ i = 0, \cdots, N_S + N_T\}$ denotes the resource set of the system.

Other basic elements of the RNS are the control points where the trains are authorized to enter a generic node by the real-time traffic controller. In addition, a path (or route) π_k is assigned to each train $v_k \in V$ travelling in the system: each vehicle starts its travel from a station; it reaches a destination station and finally the docking station where a new path can be assigned to it. More precisely, each path is described by the following sequence of resources that ends at the docking station: $\pi_k = (r_{k_1}, \cdots, r_{k_{N_k}}, r_0)$. The set A collects all the possible paths planned in the system.

Example 5.1. Let us consider the railway composed by five stations r_i, for $i = 1, \cdots, 5$, depicted in Figure 2.

The first station is a three track station, while the remaining ones have only two tracks, i.e., $C(r_1) = 3$ and $C(r_i) = 2$ for $i = 2, 3, 4, 5$.

All intermediate tracks r_i, for $i = 6, 9, 10$, are single tracks, apart from r_7 and r_8 that represent two track segments. ■

5.1 The CPN model of the RNS

In this paper we use CPN to model RNS. In particular, places represent resources (stations and tracks), while the firing of transitions represent the flow of vehicles into the system.

The generic place $p_i \in P$ models resource $r_i \in R$ and there is a one to one relationship between resources and places, thus in the following we always refer to P as R (and to p_i as r_i). Moreover, if there exists a link that goes from node r_h to node r_i, then in the CPN we introduce a transition t_j such that $t_j \in r_h{}^\bullet$ and $t_j \in {}^\bullet r_i$ [3]. Note that each transition represents a control point where the controller can stop the trains or can authorize a train to move on. Thus all transitions in the CPN model are assumed to be both controllable and observable.

[3] Given a node $x \in P \cup T$ we denote as ${}^\bullet x$ and $x^\bullet$ the preset and the postset of x, respectively.

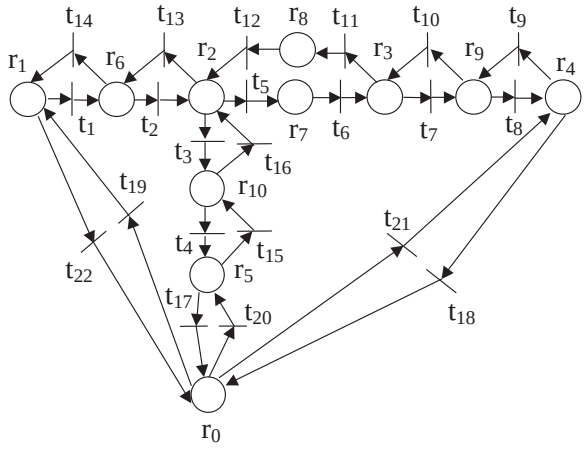

Figure 3: The CPN model of the RNS in Figure 2.

A colored token in a place represents a vehicle in a resource. The color of each token specifies the vehicle v_k or, equivalently, the routing π_k assigned to the train. As an example, $\pi_k = (r_h, \cdots, r_q, r_0)$ can be the path (i.e. the sequence of resources) assigned to the train v_k. Hence, for each $r_i \in R$ we have $Co(r_i) = \{\pi_k \mid \pi_k \text{ contains } r_i\}$, and for each $t_j \in T$ such that $t_j \in r_h{}^\bullet$ and $t_j \in {}^\bullet r_i$, we have $Co(t_j) = \{\pi_k \mid \pi_k \text{ contains } r_h \text{ and } r_i \text{ in strict succession order}\}$.

Example 5.2. Let us consider the RNS described in Example 5.1. The corresponding CPN model is reported in Figure 3 where place r_0 represents the docking station.

Let us assume that four trains are travelling in the system, namely, v_1, v_2, v_3 and v_4. Moreover, let $\pi_1 = (r_1, r_6, r_2, r_7, r_3, r_9, r_4, r_0)$, $\pi_2 = (r_4, r_9, r_3, r_8, r_2, r_6, r_1, r_0)$, $\pi_3 = (r_1, r_6, r_2, r_{10}, r_5, r_0)$, $\pi_4 = (r_5, r_{10}, r_2, r_6, r_1, r_0)$.

Therefore, by definition $Co(r_1) = \{\pi_1, \pi_2, \pi_3, \pi_4\}$ for $i = 0, 1, 2, 6$, $Co(r_i) = \{\pi_1, \pi_2\}$ for $i = 3, 4, 9$, $Co(r_7) = \{\pi_1\}$, $Co(r_8) = \{\pi_2\}$, $Co(r_i) = \{\pi_3, \pi_4\}$ for $i = 5, 10$, and $Co(t_j) = \{\pi_1, \pi_3\}$ for $j = 1, 2, 19$, $Co(t_j) = \{\pi_2, \pi_4\}$ for $j = 13, 14, 22$, $Co(t_j) = \{\pi_3\}$ for $j = 3, 4, 17$, $Co(t_j) = \{\pi_4\}$ for $j = 15, 16, 20$, $Co(t_j) = \{\pi_1\}$ for $j = 5, 6, 7, 8, 18$, $Co(t_j) = \{\pi_2\}$ for $j = 9, 10, 11, 12, 21$.

The pre and post-incidence matrices can be easily deduced by looking at the structure of the net and at the above paths definition. As an example

$$\boldsymbol{Pre}(r_1, t_1) = \boldsymbol{Post}(r_6, t_{13}) = \begin{array}{c} \begin{matrix} \pi_1 & \pi_3 \end{matrix} \\ \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \end{array} \begin{matrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \end{matrix}$$

Now, let us assume that initially trains v_1 and v_3 are in r_1, train v_2 is in r_4 and train v_4 is in

r_5. Thus the CPN system is initially at marking

$$\boldsymbol{M}_{p,0} = \begin{bmatrix} \boldsymbol{m}_{0,0} \\ \boldsymbol{m}_{1,0} \\ \boldsymbol{m}_{2,0} \\ \boldsymbol{m}_{3,0} \\ \boldsymbol{m}_{4,0} \\ \boldsymbol{m}_{5,0} \\ \boldsymbol{m}_{6,0} \\ \vdots \\ \boldsymbol{m}_{10,0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\varepsilon} \\ 1 \otimes \pi_1 + 1 \otimes \pi_3 \\ \boldsymbol{\varepsilon} \\ \boldsymbol{\varepsilon} \\ 1 \otimes \pi_2 \\ 1 \otimes \pi_4 \\ \boldsymbol{\varepsilon} \\ \vdots \\ \boldsymbol{\varepsilon} \end{bmatrix}.$$

Using the matrix notation, each term $\boldsymbol{m}_{i,0}$ may be written as a column vector of dimension $|Co(r_i)|$. As an example,

$$\boldsymbol{m}_{1,0} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{matrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \end{matrix}, \quad \boldsymbol{m}_{4,0} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{matrix} \pi_1 \\ \pi_2 \end{matrix}$$

■

5.2 Collision prevention constraints

In this paper we deal with the *real-time traffic control* of RNS whose task is that of authorizing the movement of the trains and imposing safety constraints. Such a control can be applied to railway tracks and to station tracks, and its main goal is that of avoiding *collisions* and *deadlock* in all subsystems.

To ensure that each resource does not accommodate a number of trains that is greater than the corresponding capacity, we have to introduce appropriate *collision prevention constraints*. More precisely, for all $r_i \in R \setminus \{r_0\}$, with $i = 1, \cdots, m$, we have to impose that

$$\sum_{h=1}^{u_i} m_i(\pi_{j_h}) \leq C(r_i) \tag{5}$$

where $Co(r_i) = \{\pi_{j_1}, \cdots, \pi_{j_{u_i}}\}$ and $u_i = |Co(r_i)|$.

The capacity constraints may be rewritten in terms of a single GMEC $(\boldsymbol{W}, \boldsymbol{k})$, that we call *capacity GMEC*. The capacity GMEC will have as color set $D = \{z_1, \cdots, z_m\}$ because we need

m capacity constraints, and is defined as follows:

$$\boldsymbol{W} = \left[\begin{array}{cccc} \boldsymbol{w}_0 & \boldsymbol{w}_1 & \cdots & \boldsymbol{w}_m \end{array} \right],$$

$$\boldsymbol{w}_0 = \boldsymbol{\varepsilon}$$

$$\boldsymbol{w}_i = \begin{array}{c} \begin{array}{ccc} \pi_{j_1} & \cdots & \pi_{j_{u_i}} \end{array} \\ \left[\begin{array}{ccc} 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{array} \right] \end{array} \begin{array}{c} z_1 \\ \vdots \\ z_{i-1} \\ z_i \\ z_{i+1} \\ \vdots \\ z_m \end{array} \qquad i = 1, \cdots, m \tag{6}$$

$$\boldsymbol{k} = \left[\begin{array}{cccc} C(r_1) & \cdots & C(r_m) \end{array} \right]^T \in \mathcal{Z}(D).$$

The incidence matrix of the monitor place is equal to

$$\boldsymbol{C}_c = -\boldsymbol{W} \circ \boldsymbol{C}_p$$

where $\boldsymbol{C}_p$ is the incidence matrix of the open loop net. The incidence matrix of p_c has the following structure,

$$\boldsymbol{C}_c = \left[\begin{array}{cccc} \boldsymbol{C}_c(p_c, t_1) & \cdots & \boldsymbol{C}_c(p_c, t_{15}) \end{array} \right].$$

As an example,

$$\boldsymbol{C}_c(p_c, t_1) = \begin{array}{c} \begin{array}{cc} \pi_1 & \pi_3 \end{array} \\ \left[\begin{array}{cc} 1 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -1 & -1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array} \right] \end{array} \begin{array}{c} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \\ z_8 \\ z_9 \\ z_{10} \end{array}$$

while all the other matrices $\boldsymbol{C}_c(p_c, t_j)$, $j = 2, \cdots, 15$, are omitted here for sake of brevity.

Finally, the monitor place p_c is initialized at marking

$$\boldsymbol{m}_{c,0} = \left[\begin{array}{cccccccccc} 1 & 2 & 2 & 1 & 1 & 1 & 2 & 2 & 1 & 1 \end{array} \right]^T.$$

6 Deadlock prevention policy

The design of a deadlock prevention policy is not so easy and is the object of this section. In particular, here we show that also deadlock prevention constraints can be written in the form

of a colored GMEC.

However, in order to derive an appropriate deadlock prevention policy we first need to provide some background on digraph theory. Then we show how some theoretical results firstly obtained in the context of deadlock avoidance in Automated Manufacturing Systems [7, 8], can be used here to derive a deadlock prevention policy.

6.1 Basic definitions

A *digraph* is a couple $D = (N, E)$ where $N = \{\nu_1, \nu_2, \ldots, \nu_n\}$ is the set of vertices and $E \subseteq N \times N$ is the set of edges [14].

A *path* is a subdigraph of D composed by an alternating sequence of distinct vertices and arcs. If D contains a path from ν_i to ν_j, then ν_j is said *reachable* from ν_i. Moreover, if ν_j is reachable from ν_i and ν_i is reachable from ν_j, then the two vertices are said *mutually reachable*. A *cycle* of D is a nontrivial path in which all vertices are distinct except the first and the last one[4].

A subdigraph $D_\mu = (N_\mu, E_\mu)$ of D is called *strong* if every two vertices of N_μ are mutually reachable. Finally, a *strong component* of D is a maximal strong subdigraph, i.e., a strong subdigraph that is not contained in any other strong subdigraph of D.

6.2 Deadlock characterization

Given a CPN describing a RNS, a deadlock corresponds to a marking from which a set of enabled transitions in the plant are indefinitely disabled by the capacity GMEC: such a marking is said *deadlock marking*.

In this section we establish some necessary and sufficient conditions for deadlock occurrence based on the analysis of the digraphs associated to a CPN. Since in this paper the main interest is on RNS, for clearity of explanation, results will be presented with reference to a RNS, thus we will talk of places as resources, trains as colored tokens, etc. Note however that these results can be easily generalized to any finite colored state machine with capacity constraints and a resource (place) that can accomodate all colored users (tokens) in the system.

Let us first introduce a relation that is based on the order in which the resources in a RNS are used.

Definition 6.1. A resource r_j immediately follows a resource r_i wrt path π_k if $\pi_k = (\cdots, r_i, r_j \cdots)$. This is also denoted $r_i \rhd_k r_j$. ■

We can now define two digraphs associated to a RNS represented by a CPN.

Definition 6.2. Given a RNS represented by a CPN $N_p = (R, T, Co, \boldsymbol{Pre}, \boldsymbol{Post})$ we may associate to it two main digraphs.

- The *route digraph* $D_R = (N_R, E_R)$ describes the paths of all the trains travelling in the system. Each vertex in this graph represents a resource, i.e., $N_R = R$ while $E_R = \{e_{ij} \mid (\exists \pi_k \in A)\ r_i \rhd_k r_j\}$, i.e., an edge $e_{i,j}$ belongs to E_R if there exists a path where r_j follows r_i.

[4]What we call *cycle* is sometimes called *elementary cycle*.

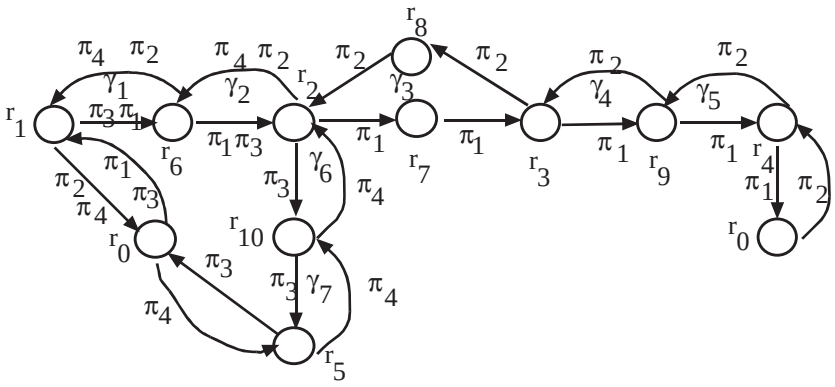

Figure 4: Digraph D_R for Example 6.3.

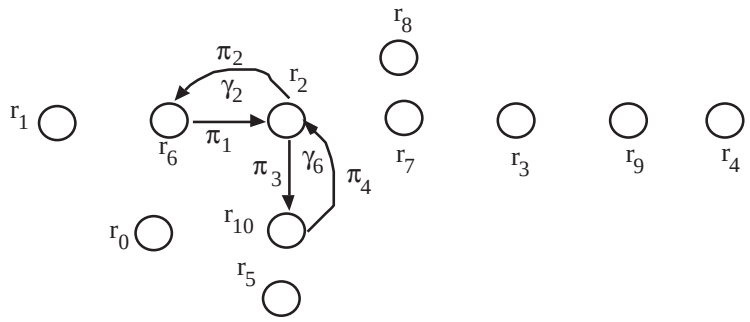

Figure 5: Digraph $D_T(\boldsymbol{M}_p)$ for Example 6.3.

- The *transition digraph* $D_T(\boldsymbol{M}_p) = (N_R, E_T(\boldsymbol{M}_p))$, describes the interactions between trains and resources when the actual marking is $\boldsymbol{M}_p$. Each vertex in this graph still represents a resource as in the route digraph, i.e., $N_R = R$, while

$$E_T(\boldsymbol{M}_p) = \{e_{ij} \mid (\exists \pi_k \in A)\boldsymbol{m}_i \geq 1 \otimes \pi_k,\ r_i \rhd_k r_j\},$$

  i.e., an edge $e_{i,j}$ belongs to $E_T(\boldsymbol{M}_p)$ if there exists a train in resource r_i at marking $\boldsymbol{M}_p$ and r_j is the next resource the train has to acquire. ■

Obviously, the arc set of the transition digraph changes as the marking is updated.

Example 6.3. Figure 4 shows digraph D_R corresponding to the system and the CPN described in Examples 5.1 and 5.2. Each edge of D_R is labelled with the name of the path to which it correspond. Moreover, for sake of simplicity, the node r_0 is repeated in Figure 4.

Assume that the four trains travelling in the system are in these positions: v_1 in r_6, v_2 and v_3 in r_2, and v_4 in r_{10}. Hence, the CPN is at marking $\boldsymbol{M}_p$ equal to $\boldsymbol{m}_6 = 1 \otimes \pi_1$, $\boldsymbol{m}_2 = 1 \otimes \pi_2 + 1 \otimes \pi_3$, $\boldsymbol{m}_{10} = 1 \otimes \pi_4$, $\boldsymbol{m}_i = \boldsymbol{\varepsilon}$ elsewhere.

The corresponding transition digraph $D_T(\boldsymbol{M}_p)$ is shown in Figure 5. ■

To characterize deadlock markings we also need the following definition.

Definition 6.4. A strong component $D_\mu = (N_\mu, E_\mu)$ of $D_T(\boldsymbol{M}_p)$ is called a *Maximal-weight Zero-outdegree Strong Component* (MZSC for brevity) if the following properties hold true:

(a) *Maximal-weight*: all the resources from N_μ are busy, i.e., the number of tokens in each place r_i is equal to the maximal capacity of the place: $|\boldsymbol{m}_i| = C(r_i)$.

(b) *Zero-outdegree*: all the edges of $D_T(\boldsymbol{M}_p)$ outgoing from vertices of N_μ belong to E_μ.

■

Remark 6.5. Note that the node r_0 corresponding to the docking station can not be in a MZSC because it has an infinite capacity and all cycles containing it may be disregarded for deadlock analysis.

It is possible to give a simple characterization of an MZSC in terms of resources allocated to the trains at a given marking.

Definition 6.6. Given a strong subdigraph $D_\mu = (N_\mu, E_\mu)$ of D_R and a marking $\boldsymbol{M}_p$, we denote the set of trains that occupy a resource of N_μ and require a resource of N_μ by an edge of E_μ at the next step as

$$V(\boldsymbol{M}_p)_\mu = \{v_k \in V \mid (\exists r_i, r_j \in N_\mu)\ \boldsymbol{m}_i \geq 1 \otimes \pi_k, \\ r_i \rhd_k r_j,\ e_{i,j} \in E_\mu\}.$$

■

Definition 6.7. Let $D_\mu = (N_\mu, E_\mu)$ be a strong subdigraph of D_R. We denote

$$C(N_\mu) = \sum_{r_i \in N_\mu} C(r_i)$$

the sum of the capacities of the resources in N_μ. ■

The following result holds.

Proposition 6.8. A necessary and sufficient condition for a strong subdigraph $D_\mu = (N_\mu, E_\mu)$ of D_R to be an MZSC in $D_T(\boldsymbol{M}_p)$ is that $|V(\boldsymbol{M}_p)_\mu| = C(N_\mu)$.

Proof. Sufficiency derives directly from Definitions 6.4 and 6.6.

To prove necessity we observe that if D_μ is a MZSC then (by the maximal-weight condition) the number of tokens in N_μ at $\boldsymbol{M}_p$ is equal to $\sum_{r_i \in N_\mu} |\boldsymbol{m}_i| = \sum_{r_i \in N_\mu} C(r_i) = C(N_\mu)$. Furthermore, each of these tokens correspond to a train in the set of resources N_μ that (by the zero-outdegree condition) requires at the next step a resource in N_μ, i.e., $C(N_\mu) = |V(\boldsymbol{M}_p)_\mu|$. □

In [7] and [8] necessary and sufficient conditions for deadlock occurrence have been characterized in terms of digraph analysis.

Proposition 6.9. A marking $\boldsymbol{M} = [\boldsymbol{M}_p^T\ \boldsymbol{m}_c]^T$ is a deadlock marking for a CPN with capacity constraint iff there exists at least one MZSC in $D_T(\boldsymbol{M}_p)$.

Proof. The statement is a slightly different formulation of Theorem 1 from [7] in terms of net marking rather than system state. As such, it applies to CPN modelling RNS. □

From Propositions 6.8 and 6.9, the following corollary is derived.

Corollary 6.10. The marking $\boldsymbol{M} = [\boldsymbol{M}_p^T \ \boldsymbol{m}_c]^T$ is a deadlock marking for a CPN with capacity constraints iff there exists a strong component $D_\mu = (N_\mu, E_\mu)$ of D_R such that $|V(\boldsymbol{M}_p)_\mu| = C(N_\mu)$. □

Example 6.11. Let us consider again Example 6.3 and the transition digraph $D_T(\boldsymbol{M}_p)$ in Figure 5 corresponding to the defined marking $\boldsymbol{M}_p$. It is easy to verify that the strong component $D_\mu = \gamma_2 \cup \gamma_6 = (\{r_6, r_2, r_{10}\}, \{e_{6,2}, e_{2,10}, e_{10,2}, e_{2,6}\})$ of D_R is an MZSC in $D_T(\boldsymbol{M}_p)$. Moreover, we obtain: $V(\boldsymbol{M}_p)_\mu = \{v_1, v_2, v_3, v_4\}$, and $|V(\boldsymbol{M}_p)_\mu| = C(N_\mu) = 4$. ■

6.3 Second level deadlocks

By imposing constraints of the form $|V(\boldsymbol{M}_p)_\mu| \leq C(N_\mu) - 1$ we can prevent any strong component of D_R from becoming an MZSC in the transition digraph.

However, avoiding a deadlock marking is not sufficient to guarantee the liveness of the CPN. Indeed, it is possible that some critical states are reached that are not deadlocks, but they necessary evolve to a deadlock marking in the next step: these states are called *Second Level Deadlocks* (SLD) [7]. Clearly, if a SLD marking is reached, then a controller that has been designed to prevent reaching a deadlock marking for the original net will create a new deadlock marking.

We discuss in this subsection how it may be possible to also prevent a SLD.

A SLD can be characterized in terms of a particular interaction among the cycles of D_R that can be represented by a new digraph.

Definition 6.12. Given a CPN with route digraph $D_R = (N_R, E_R)$ we define the *second level digraph* $D_R^2 = (N^2, E_R^2)$ such that:

- the set of vertices $N^2 = \{\gamma_1, \gamma_2, \cdots, \gamma_N\}$ is equal to the set of cycles of D_R that do not contain the dummy node[5] r_0;
- an edge $e_{u,s}$ belongs to E_R^2 if:
  - i) γ_u and γ_s have only one vertex in common (say r_j) with capacity $C(r_j) = 1$;
  - ii) there exists a path $\pi_k \in A$ such that $r_i \rhd_k r_j \rhd_k r_h$ requiring resources r_i, r_j, r_h in strict order of succession, with $e_{i,j} \in \gamma_u$ and $e_{j,h} \in \gamma_s$. ■

Now let γ_u^2 be a cycle in D_R^2 (second level cycle). From the previous definitions it follows that a subset of cycles of D_R (say Γ_u) is associated with vertices of γ_u^2. The set $\Gamma_u = (N_{\Gamma_u}, E_{\Gamma_u})$ is a strong subdigraph of D_R and its capacity is

$$C(N_{\Gamma_u}) = C(\gamma_u^2) = \sum_{r \in \{\gamma \ | \ \gamma \in \Gamma_u\}} C(r)$$

i.e., it is equal to the sum of the capacities of all the resources in Γ_u.

[5]The dummy node r_0 has infinity capacity and all cycles containing it may be disregarded for deadlock analysis as mentioned in remark 6.5.

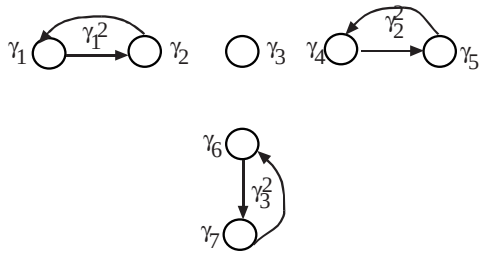

Figure 6: Digraph D_R^2 obtained from digraph D_R in Figure 4.

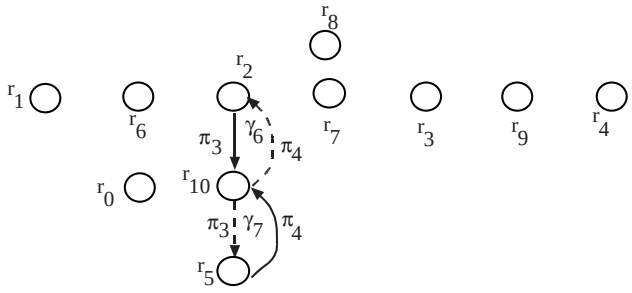

Figure 7: Digraph $D_T(\boldsymbol{M}_p)$ for Example 6.14.

Finally, let Γ^2 indicate the subset of cycles of D_R^2, enjoying the following property: $\gamma_u^2 \in \Gamma^2$ iff the corresponding set Γ_u collects cycles that are all disjoint except for one vertex of unit capacity, common to all of them.

For each $\gamma_u^2 \in \Gamma^2$ and for each marking $\boldsymbol{M}_p$ we introduce the following set: $V(\boldsymbol{M}_p)_{\gamma_u^2} = \{v_k \in V \mid (\exists\, r_i \in N_{\Gamma_\mu})\ \boldsymbol{m}_i \geq 1 \otimes \pi_k\}$.

As shown by the following proposition easily derived from the results proved in [7], the set Γ^2 plays an important role in defining SLD conditions.

Proposition 6.13. If $\boldsymbol{M} = [\boldsymbol{M}_p^T\ \boldsymbol{m}_c]^T$ is a SLD marking for a CPN with capacity constraint, then there exists in D_R^2 a *second level cycle* $\gamma_u^2 \in \Gamma^2$ such that: $|V(\boldsymbol{M}_p)_{\gamma_u^2}| = C(\gamma_u^2) - 1$. □

Example 6.14. Considering the cycles of D_R depicted in Figure 4, the second level cycles of D_R^2 are derived and shown in Figure 6.

We suppose there are four trains in the system so that the CPN is at marking $\boldsymbol{M}_p$ where $\boldsymbol{m}_2 = 2 \otimes \pi_3$, $\boldsymbol{m}_5 = 2 \otimes \pi_4$, and $\boldsymbol{m}_i = \boldsymbol{\varepsilon}$ elsewhere. The solid lines in Figure 7 depicts the transition digraph $D_T(\boldsymbol{M}_p)$. For convenience, Figure 7 also depicts the second transitions (dashed lines) in the residual paths of the trains.

The described marking exhibits a SLD for the CPN. Indeed, the second level cycle $\gamma_3^2 \in \Gamma^2$ is in second level deadlock condition, where γ_3^2 corresponds to the cycle set $\Gamma_u = \gamma_6 \cup \gamma_7 = (\{r_2, r_{10}, r_5\}, \{e_{2,10},\ e_{10,5}, e_{5,10}, e_{10,2}\})$. Hence, the necessary condition of Proposition 6.13 is verified, i.e., $|V(\boldsymbol{M}_p)_{\gamma_3^2}| = C(\gamma_3^2) - 1 = 4$. ■

6.4 Deadlock prevention

A prevention policy is complicated for the following reasons. As already discussed above it is quite easy to characterize the deadlock states: its detection is performed in a polynomial complexity on the number of nodes of the digraph [7]. A naif approach would be that of designing a control policy that forbids such a deadlock states. However, this deadlock prevention policy may introduce further second order deadlocks that require the introduction of a second order policy that in its term introduces third order deadlock, and so on. The computational complexity of this procedure is known to be in the class of NP-complete problems [28]. In this paper we propose a sub-optimal but efficient procedure that studies first and second order deadlock only and designs a deadlock prevention policy that, by satisfying an additional constraint, ensures the deadlock freeness of the system. Thus, we trade-off permissiveness for computational tractability.

In this subsection we first discuss in detail how the proposed prevention policy can be computed. Then we prove that this policy ensures deadlock freeness.

Corollary 6.10 and Proposition 6.13 establish the deadlock and SLD prevention conditions on the marking of the CPN. To obtain a direct relation among $|V(\boldsymbol{M}_p)_\mu|$, where $D_\mu = (N_\mu, E_\mu)$ is a strong subdigraph of D_R, and the marking of the CPN, the following index set is defined for each $r_i \in N_\mu$:

$$H_\mu(r_i) = \{h \mid \quad a_{i,h} \in Co(r_i) \cap Co(r_k) \text{ with } e_{i,k} \in E_\mu \text{ and } r_k \in N_\mu\}.$$

A controller will prevent reaching a deadlock or a SLD marking if it can enforce the following *deadlock prevention GMEC*:

- for each strong subdigraph D_μ of D_R

$$|V(\boldsymbol{M}_p)_\mu| = \sum_{r_i \in N_\mu} \sum_{h \in H_\mu(r_i)} \boldsymbol{m}_i(h) \leq C(N_\mu) - 1 \tag{7}$$

- for each $\gamma_u^2 \in \Gamma^2$ of D_R^2

$$|V(\boldsymbol{M}_p)_{\gamma_u^2}| = \sum_{r_i \in N_{\Gamma_u}} |\boldsymbol{m}_i| \leq C(\gamma_u^2) - 2. \tag{8}$$

The GMEC given by Equations (7) and (8) restrict the reachability set of the closed loop plant, to avoid deadlock and SLD. However, the imposed GMEC can eventually lead to a situation similar to a deadlock, which is known as *restricted deadlock* (RD).More precisely, in a restricted condition a set of transitions keeps on remaining indefinitely inhibited by the imposed constraints. Such a situation is determined when two or more constraints simultaneously inhibit a set of transitions. The following example clarifies the situation.

Example 6.15. Assume that five trains travel in the system described in Examples 5.1 and 5.2. Let us suppose that the CPN is at marking $\boldsymbol{M}_p$ such that $\boldsymbol{m}_1 = 2 \otimes \pi_1$, $\boldsymbol{m}_2 = \boldsymbol{m}_6 = 1 \otimes \pi_3$, $\boldsymbol{m}_{10} = 1 \otimes \pi_4$. Figure 8 depicts the corresponding transition digraph $D_T(\boldsymbol{M})$. We note that transitions t_1 and t_3 of the CPN are color enabled but they are inhibited by the capacity constraints. Moreover, transition t_2 is inhibited by the constraint $|V(\boldsymbol{M}_p)_{\gamma_6}| \leq 2$ and transition

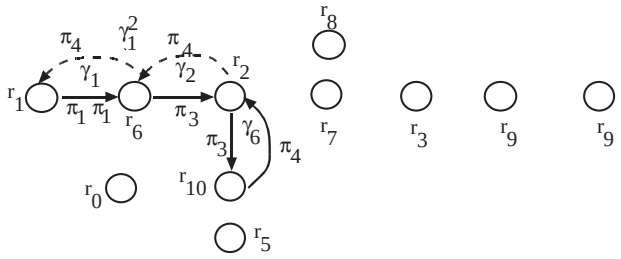

Figure 8: Digraph $D_T(\boldsymbol{M})$ for Example 6.15.

t_{16} is inhibited by the constraint $|V(\boldsymbol{M}_p)_{\gamma_1^2}| \leq 4$. Hence, all the enabled transitions are inhibited by the imposed constraints and a restricted deadlock occurs. ■

Definition 6.16. Let $\boldsymbol{M} = [\boldsymbol{M}_p^T \ \boldsymbol{m}_c]^T$ be not a deadlock marking and a SLD marking for the CPN with capacity constraints and deadlock prevention GMEC. The marking $\boldsymbol{M}$ is a RD marking if there exists a set of color enabled transitions that keep on remaining inhibited by the imposed constraints. ■

Remark 6.17. We recall that if the system is constrained by capacity constraints and deadlock prevention constraints of type (7), the system is in a restricted deadlock condition if and only if it is in a second level deadlock state [7, 8]. ■

The following proposition establishes the conditions that must be verified to obtain a RD free closed loop system.

Proposition 6.18. If $\boldsymbol{M} = [\boldsymbol{M}_p^T \ \boldsymbol{m}_c]^T$ is a RD marking for a CPN with capacity constraints and deadlock prevention GMEC, then one of the following two conditions are verified:

a) there exist at least two cycle sets $\Gamma_1 = (N_{\Gamma_1}, E_{\Gamma_1})$ and $\Gamma_2 = (N_{\Gamma_2}, E_{\Gamma_2})$ of D_R corresponding to the second level cycles $\gamma_1^2, \gamma_2^2 \in \Gamma^2$ such that $|V(\boldsymbol{M}_p)_{\gamma_1^2}| = C(N_{\Gamma_1}) - 2$ and $|V(\boldsymbol{M}_p)_{\gamma_2^2}| = C(N_{\Gamma_2}) - 2$;

b) there exists at least a cycle set $\Gamma_1 = (N_{\Gamma_1}, E_{\Gamma_1})$ of D_R corresponding to $\gamma_1^2 \in \Gamma^2$ and a cycle $\gamma_1 = (N_{\gamma_1}, E_{\gamma_1})$ of D_R such that $|V(\boldsymbol{M}_p)_{\gamma_1^2}| = C(N_{\Gamma_1}) - 2$ and $|V(\boldsymbol{M}_p)_{\gamma_1}| = C(N_{\gamma_1}) - 1$.

Proof. Let $\boldsymbol{M}_p$ be a RD marking of the closed loop system. The restricted deadlock is not determined by the GMEC in Equation (7) because $\boldsymbol{M}_p$ is not a SLD marking (see Remark 6.17). Hence, one of the following conditions is verified.

1) There exist at least two constraints defined by Equation (8) that determine the RD marking, say $|V(\boldsymbol{M}_p)_{\gamma_1^2}| \leq C(N_{\Gamma_1}) - 2$ and $|V(\boldsymbol{M}_p)_{\gamma_2^2}| \leq C(N_{\Gamma_2}) - 2$. More precisely there is a transition $t_j \in T$ that is inhibited by the constraint $|V(\boldsymbol{M}_p)_{\gamma_1^2|}| \leq C(N_{\Gamma_1}) - 2$ and a transition $t_{j'} \in T$ that is inhibited by the constraint $|V(\boldsymbol{M}_p)_{\gamma_2^2}| < C(N_{\Gamma_2}) - 2$. Hence, there exists an edge $e_{v,q} \in E_R$, corresponding to transition t_j, such that $r_v \notin N_{\Gamma_1}$, $r_q \in N_{\Gamma_1}$, and $|V(\boldsymbol{M}_p)_{\gamma_1^2}| = C(N_{\gamma_1}) - 2$. Analogously, there exists an edge $e_{i,m} \in E_R$, corresponding to transition $t_{j'}$, such that $r_i \notin N_{\Gamma_2}$, $r_m \in N_{\Gamma_2}$, and $|V(\boldsymbol{M}_p)_{\gamma_2^2}| = C(N_{\Gamma_2}) - 2$. This proves condition a).

b) There exists at least one constraint defined by Equation (8) (say $|V(\boldsymbol{M}_p)_{\gamma_1^2}| \leq C(N_{\Gamma_1}) - 2$) and one constraint of type (7) (say $|V(\boldsymbol{M}_p)_{\gamma_1}| \leq C(N_{\Gamma_1}) - 1$) that cause the RD marking. More

precisely there is a transition $t_j \in T$ that is inhibited by the constraint $|V(\boldsymbol{M}_p)_{\gamma_1^2}| \leq C(N_{\Gamma_1}) - 2$ and a transition $t_{j'} \in T$ that is inhibited by the constraint $|V(\boldsymbol{M}_p)_{\gamma_1}| \leq C(N_{\Gamma_1}) - 1$. Hence, there exists an edge $e_{v,q} \in E_R$, corresponding to transition t_j, such that $r_v \notin N_{\Gamma_1}$, $r_q \in N_{\Gamma_1}$, and $|V(\boldsymbol{M}_p)_{\gamma_1^2}| = C(N_{\Gamma_1}) - 2$. Analogously, there exists $e_{i,m} \in E_R$, corresponding to transition $t_{j'}$, such that $r_i \notin N_{\gamma_1}$, $r_m \in N_{\gamma_1}$, and $|V(\boldsymbol{M}_p)_{\gamma_1}| = C(N_{\gamma_1}) - 1$. So necessary condition b) is proved. □

Now, let us define the following parameters:

- $C_{01} = \min\{C(N_{\Gamma_i}) - 2 + C(N_{\Gamma_j}) - 2, \quad \text{for each } \gamma_i^2, \gamma_j^2 \in \Gamma^2 \text{ with } i \neq j\}$
- $C_{02} = \min\{C(N_{\Gamma_i}) - 2 + C(N_{\gamma_j}) - 1, \quad \text{for each } \gamma_i^2 \in \Gamma^2 \text{ and for each } \gamma_j \text{ of } D_R\}$
- $C_0 = \min\{C_{01}, C_{02}\}$

Corollary 6.19. If $|\boldsymbol{M}_{p,0}| < C_0$, the closed loop system with the monitors that enforce the deadlock prevention GMEC in Equations (7) and (8) is deadlock free.

Proof. Follows from Proposition 6.18, because necessary conditions a) and b) can not be verified by all markings $\boldsymbol{M}_p$ reachable from $\boldsymbol{M}_{p,0}$.

The previous corollary only gives a sufficient condition for deadlock prevention and one may wonder how conservative the proposed policy is. The value of C_0, that constrains the number of trains admitted in the system depends on the cycle and strong subdigraph capacities. Qualitatively, we may say that if the station and tracks have sufficiently large capacities with respect with the number of trains in service, as it is often the cases in real applications, the obtained value of C_0 is high enough to provide a reasonably good policy.

6.5 Computational complexity

Let us now discuss the computational complexity of the proposed approach. We first point out that the deadlock prevention policy requires no on-line but just off-line computations. More precisely, to establish the deadlock prevention GMEC it is necessary to compute off-line the cycles of D_R and of digraph D_R^2. Technical literature [29] provide algorithms for generating cycles of D_R in $\mathcal{O}\{[(\text{Card}(N_R) + \text{Card}(E_R)](c_1 + 1)\}$ time, where c_1 represents the number of cycles of D_R. Building D_R^2 can be performed in $\mathcal{O}[(c_1)^2 L]$ operations, where L indicates the sum of the lengths of all the possible paths (i.e., the sum of resources appearing in all the paths, counting repetitions). Moreover, generating the cycles of D_R^2 and characterizing Γ^2 need $\mathcal{O}\{[c_1 + \text{Card}(E_R^2)](c_2 + 1)\}$ and $\mathcal{O}(c_1 c_2)$ operations, respectively, where c_2 indicates the number of second level cycles. To sum up, the complexity of these algorithms depends quadratically on the number of cycles of D_R and D_R^2 that may be very high (the maximum number of cycles of a digraph is exponential in the number of nodes). However, the considered digraphs are not complete and the algorithms are employed once, before the proper real time control. Hence, the presented strategy can be applied to large systems. Note however, that large and complex RNS are not controlled by centralized schemes but are decoupled in subsystems that are governed separately and independently.

6.6 Deadlock prevention constrains in RNS

Let us consider again the CPN in Figure 3.

To obtain the deadlock prevention GMEC, we have to determine the sets $V(\boldsymbol{M}_p)_{\gamma_v}$ associated with cycle γ_v with $v = 1, \cdots, 7$, the sets $V(\boldsymbol{M}_p)_{\mu_v}$ associated with each strong subdigraph D_μ of D_R (see Figure 4) and the sets $V(\boldsymbol{M}_p)_{\gamma_u^2}$ with $u = 1, \cdots, 3$ associated with the second level cycles $\gamma_u^2 \in \Gamma^2$ of D_R^2 with $u = 1, 2, 3$.

Moreover, we obtain $C_{01} = 6$, $C_{02} = 5$, thus $C_0 = \min\{6, 5\} = 5$.

Considering the previously defined sets, the following conditions are imposed by the prevention policy:

$$
\begin{cases}
|V(\boldsymbol{M}_p)_{\gamma_1}| \leq 3 & m_1(\pi_1) + m_1(\pi_3) + \\
 & m_6(\pi_2) + m_6(\pi_4) \leq 3 \\
|V(\boldsymbol{M}_p)_{\gamma_2}| \leq 2 & m_2(\pi_2) + m_2(\pi_4) + \\
 & m_6(\pi_1) + m_6(\pi_3) \leq 2 \\
|V(\boldsymbol{M}_p)_{\gamma_4}| \leq 2 & m_3(\pi_1) + m_9(\pi_2) \leq 2 \\
|V(\boldsymbol{M}_p)_{\gamma_5}| \leq 2 & m_4(\pi_2) + m_9(\pi_1) \leq 2 \\
|V(\boldsymbol{M}_p)_{\gamma_6}| \leq 2 & m_2(\pi_3) + m_{10}(\pi_4) \leq 2 \\
|V(\boldsymbol{M}_p)_{\gamma_7}| \leq 2 & m_5(\pi_4) + m_{10}(\pi_3) \leq 2 \\
|V(\boldsymbol{M}_p)_{\gamma_2 \cup \gamma_6}| \leq 3 & m_2(\pi_2) + + m_2(\pi_3) + \\
 & m_2(\pi_4) + m_6(\pi_1) + \\
 & m_6(\pi_3) + m_{10}(\pi_4) \leq 3 \\
|V(\boldsymbol{M}_p)_{\gamma_2^2}| \leq 3 & m_3(\pi_1) + m_3(\pi_2) + \\
 & m_4(\pi_1) + m_4(\pi_2) \\
 & + m_9(\pi_1) + m_9(\pi_2) \leq 3 \\
|V(\boldsymbol{M}_p)_{\gamma_3^2}| \leq 3 & m_2(\pi_1) + m_2(\pi_2) + \\
 & m_2(\pi_3) + m_2(\pi_4) + \\
 & m_5(\pi_3) + m_5(\pi_4) + \\
 & m_{10}(\pi_3) + m_{10}(\pi_4) \leq 3
\end{cases}
$$

Since the initial marking is such that $|\boldsymbol{M}_{p,0}| < C_0$, the resulting closed loop system is deadlock and RD free.

Note that if we suppose four trains in the system ($|\boldsymbol{M}_{p,0}| < 5$), some constraints are always verified. For example, $|V(\boldsymbol{M}_p)_{\gamma_3}| \leq 5$, $|V(\boldsymbol{M}_p)_{\gamma_2 \cup \gamma_3}| \leq 6$, $|V(\boldsymbol{M}_p)_{\gamma_3 \cup \gamma_6}| \leq 6$, $|V(\boldsymbol{M}_p)_{\gamma_1^2}| \leq 4$. Obviously, the deadlock prevention strategy can be simplified neglecting such constraints.

The above 9 constraints can be rewritten in terms of a single GMEC $(\boldsymbol{W}', \boldsymbol{k}')$, that we call *deadlock prevention GMEC*. The deadlock prevention GMEC will have as color set $D' = \{z_1', \cdots, z_9'\}$ because we have to impose 9 constraints, and is defined as follows:

$$
\boldsymbol{W}^T = \begin{bmatrix} \boldsymbol{w}_0^T & \boldsymbol{w}_1^T & \cdots & \boldsymbol{w}_{10}^T \end{bmatrix}
$$

$$
\boldsymbol{w}_0^T = \boldsymbol{\varepsilon}
$$

$$
\boldsymbol{k}^T = \begin{bmatrix} 3 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 \end{bmatrix}^T \in \mathcal{Z}(D').
$$

As an example, we report here the numerical values of $\boldsymbol{w}_2^T$ and $\boldsymbol{w}_3^T$, while the other $\boldsymbol{w}_i$'s are

omitted for sake of brevity:

$$\boldsymbol{w}_2^T = \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} \pi_1 & \pi_2 & \pi_3 & \pi_4 \end{matrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{matrix} z'_1 \\ z'_2 \\ z'_3 \\ z'_4 \\ z'_5 \\ z'_6 \\ z'_7 \\ z'_8 \\ z'_9 \end{matrix} \quad \boldsymbol{w}_3^T = \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} \pi_1 & \pi_2 \end{matrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \\ 0 & 0 \end{bmatrix} \begin{matrix} z'_1 \\ z'_2 \\ z'_3 \\ z'_4 \\ z'_5 \\ z'_6 \\ z'_7 \\ z'_8 \\ z'_9 \end{matrix}$$

Note that each matrix $\boldsymbol{w}_i^T$ has as many rows as the number of constraints (i.e., 9 rows) and as many columns as the number of colors that may be contained in place r_i.

Moreover, by looking at matrix $\boldsymbol{w}_2^T$ we may observe that its first row is null because the marking $\boldsymbol{m}_2$ is not involved in the first constraint. On the contrary, the non null elements in its second row, relative to π_2 and π_4 are due to the fact that $m_2(\pi_2)$ and $m_2(\pi_4)$ are involved in the second constraint. The value 1 is due to the fact that 1 is the associated coefficient in the corresponding linear constraint.

The incidence matrix of the monitor place is equal to

$$\boldsymbol{C}_c = -\boldsymbol{W} \circ \boldsymbol{C}_p$$

where $\boldsymbol{C}_p$ is the incidence matrix of the open loop net. The incidence matrix of p_c has the following structure,

$$\boldsymbol{C}_c = \begin{bmatrix} \boldsymbol{C}_c(p_c, t_1) & \cdots & \boldsymbol{C}_c(p_c, t_{15}) \end{bmatrix}.$$

As an example,

$$\boldsymbol{C}_c(p_c, t_1) = \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} \pi_1 & \pi_3 \end{matrix} \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -1 & -1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{matrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \\ z_8 \\ z_9 \\ z_{10} \end{matrix}$$

while all the other matrices $\boldsymbol{C}_c(p_c, t_j)$, $j = 2, \cdots, 15$, are omitted here for sake of brevity.

Finally, the monitor place p_c is initialized at marking

$$\boldsymbol{m}_{c,0} = \begin{bmatrix} 1 & 2 & 2 & 1 & 1 & 1 & 2 & 2 & 1 & 1 \end{bmatrix}^T.$$

A similar reasoning may be repeated to impose the deadlock prevention constraints, being in the form of colored GMEC.

7 Conclusions

The contribution of this paper is twofold.

On one side we have extended the classic PN control approach based on GMEC and monitor places to the case of CPN. A colored GMEC can express a set of linear constraints and can be enforced by a colored monitor place. We have also developed a matrix representation of multisets that is useful for the design of the monitor place.

On the other side, we provided a CPN model to describe a RNS and to derive the traffic controller. The introduced framework allowed us to define a supervisor controller guaranteeing safeness and deadlock freeness in the railway traffic control system. Starting from the analysis of deadlock on the basis of digraph tools, a deadlock prevention strategy was defined and expressed by a set of linear inequality constraints. Moreover, we shown how collision and deadlock prevention constraints can be expressed as colored GMEC and the controller can be realized by a set of monitor places.

A Appendix: Multisets

In this section we recall some notation that will be useful in the following, when formally defining the colored PN model.

Definition A.1. Let D be a set. A *multiset* (resp., *non negative multiset*) $\boldsymbol{\alpha}$ over D is defined by a mapping $\alpha : D \to \mathbb{Z}$ ($\alpha : D \to \mathbb{N}$) and is represented using a special symbol $\otimes$ as

$$\boldsymbol{\alpha} = \sum_{d \in D} \alpha(d) \otimes d$$

where the sum is limited to the elements such that $\alpha(d) \neq 0$.

Let $\mathcal{Z}(D)$ (resp., $\mathcal{N}(D)$) denote the set of all multisets (resp., non negative multisets) over D.

The multiset $\boldsymbol{\varepsilon}$ is the empty multiset such that for all $d \in D,\ \varepsilon(d) = 0$. ■

Definition A.2. Given two multisets $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathcal{Z}(D)$ and a number $a \in \mathbb{Z}$:

- The sum of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is denoted as $\boldsymbol{\gamma} = \boldsymbol{\alpha} + \boldsymbol{\beta}$ and is defined as $\forall d \in D : \gamma(d) = \alpha(d) + \beta(d)$.
- The difference of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is denoted as $\boldsymbol{\gamma} = \boldsymbol{\alpha} - \boldsymbol{\beta}$ and is defined as $\forall d \in D : \gamma(d) = \alpha(d) - \beta(d)$. Note that the difference of two non negative multisets may be negative.
- The product of $\boldsymbol{\alpha}$ and a is denoted as $\boldsymbol{\gamma} = a\ \boldsymbol{\alpha}$ and is defined as $\forall d \in D : \gamma(d) = a\ \alpha(d)$.
- We write $\boldsymbol{\alpha} \leq \boldsymbol{\beta}$ iff $\forall d \in D : \alpha(d) \leq \beta(d)$.

■

Now, given two sets D and D', let $\boldsymbol{F} : D \to \mathcal{Z}(D')$ be a function that associates to each element $d \in D$ a multiset on D':

$$\boldsymbol{F}(d) = \sum_{d' \in D'} F(d, d') \otimes d' \in \mathcal{Z}(D').$$

We can naturally extend this application to a function $\boldsymbol{F} : \mathcal{Z}(D) \to \mathcal{Z}(D')$ as follows.

Definition A.3. Given two sets D and D', a function $\boldsymbol{F} : D \to \mathcal{Z}(D')$, and a multiset $\boldsymbol{\alpha} \in \mathcal{Z}(D)$, we define

$$\boldsymbol{F} \circ \boldsymbol{\alpha} \triangleq \sum_{d \in D} \alpha(d) \boldsymbol{F}(d) = \sum_{d \in D} \sum_{d' \in D'} \alpha(d) F(d, d') \otimes d' \in \mathcal{Z}(D')$$

i.e., using the special symbol $\circ$, the linear combination with coefficients $\alpha(d)$ of the multisets $\boldsymbol{F}(d)$ over D' is denoted $\boldsymbol{F} \circ \boldsymbol{\alpha}$. ■

A simple example will help to clarify the notation.

Example A.4. Let us consider the two sets $D = \{c_1, c_2\}$ and $D' = \{z_1, z_2, z_3\}$, and the multiset $\boldsymbol{\alpha}$ over D, where $\boldsymbol{\alpha} = 2 \otimes c_1 + 3 \otimes c_2$. Let $\boldsymbol{F}(c_1) = 4 \otimes z_1 + 5 \otimes z_2 + 2 \otimes z_3$ and $\boldsymbol{F}(c_2) = 3 \otimes z_1 + 2 \otimes z_2 + 2 \otimes z_3$ be two multisets over D'. Then, by definition,

$$\begin{aligned} \boldsymbol{F} \circ \boldsymbol{\alpha} &= \textstyle\sum_{d \in \{c_1, c_2\}} \alpha(d) \boldsymbol{F}(d) \\ &= 2\boldsymbol{F}(c_1) + 3\boldsymbol{F}(c_2) \\ &= (2 \cdot 4 + 3 \cdot 3) \otimes z_1 + (2 \cdot 5 + 3 \cdot 2) \otimes z_2 + (2 \cdot 2 + 3 \cdot 2) \otimes z_3 \\ &= 17 \otimes z_1 + 16 \otimes z_2 + 10 \otimes z_3 \in \mathcal{Z}(D') \end{aligned}$$

■

We finally observe that it is possible to give a matrix representation of multisets and of functions over multisets.

Remark A.5. Given two sets D and D', let us arbitrary order their elements as follows: $D = \{d_1, \ldots, d_k\}$ and $D' = \{d'_1, \ldots, d'_{k'}\}$.

A multiset $\boldsymbol{\alpha} \in \mathcal{Z}(D)$ can be represented by a vector:

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha(d_1) \\ \alpha(d_2) \\ \vdots \\ \alpha(d_k) \end{bmatrix} \in \mathbb{Z}^k.$$

Thus, given a function $\boldsymbol{F} : D \to \mathcal{Z}(D')$ for all $d \in D$ we can write

$$\boldsymbol{F}(d) = \begin{bmatrix} F(d, d'_1) \\ F(d, d'_2) \\ \vdots \\ F(d, d'_{k'}) \end{bmatrix} \in \mathbb{Z}^{k'}.$$

while its extension $\boldsymbol{F} : \mathcal{Z}(D) \to \mathcal{Z}(D')$ can be represented by the matrix

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{F}(d_1) & \boldsymbol{F}(d_2) & \ldots & \boldsymbol{F}(d_k) \end{bmatrix} \in \mathbb{Z}^{k' \times k}$$

and finally the multiset $\boldsymbol{F} \circ \boldsymbol{\alpha}$ can be computed with the usual matrix-vector product denoted by $\cdot$, i.e.,

$$\boldsymbol{F} \circ \boldsymbol{\alpha} = \begin{bmatrix} \sum_{i=1}^{k} \alpha(d_i) F(d_i, d'_1) \\ \sum_{i=1}^{k} \alpha(d_i) F(d_i, d'_2) \\ \vdots \\ \sum_{i=1}^{k} \alpha(d_i) F(d_i, d'_{k'}) \end{bmatrix} \in \mathbb{Z}^{k'}.$$

■

Example A.6. Let us go back to the Example A.4. We can write

$$\boldsymbol{F} = \left[\begin{array}{cc} \boldsymbol{F}(c_1) & \boldsymbol{F}(c_2) \end{array} \right] = \begin{array}{c} \begin{array}{cc} c_1 & c_2 \end{array} \\ \left[\begin{array}{cc} 4 & 3 \\ 5 & 2 \\ 2 & 2 \end{array} \right] \end{array} \begin{array}{c} \\ z_1 \\ z_2 \\ z_3 \end{array}$$

and thus

$$\boldsymbol{F} \circ \boldsymbol{\alpha} = \left[\begin{array}{cc} 4 & 3 \\ 5 & 2 \\ 2 & 2 \end{array} \right] \cdot \left[\begin{array}{c} 2 \\ 3 \end{array} \right] = \left[\begin{array}{c} 17 \\ 16 \\ 10 \end{array} \right].$$

■

B Abbreviation list

- wrt: with respect to,
- PN: Petri net,
- CPN: colored Petri net,
- GMEC: generalized mutual exclusion constraint,
- RNS: railway network system,
- MZSC: maximal-weight zero-outdegree strong component,
- SLD: second level deadlock,
- RD: restricted deadlock.

C Symbol list

- $\mathcal{Z}(D)$: set of all multisets over D,
- $\mathcal{N}(D)$: set of all non negative multisets over D,
- $\boldsymbol{\varepsilon}$: empty multiset,
- Co: color function,
- $\mathcal{C}l$: set of possible colors,
- σ: firing sequence,
- $\boldsymbol{\Sigma}$: firing count vector,
- $\mathcal{M}(\boldsymbol{W}, \boldsymbol{k})$: set of legal markings defined by ($\boldsymbol{W}$, $\boldsymbol{k}$),
- p_c: monitor place,
- v_k: k-th vehicle (train),

- N_S: number of stations,
- N_T: number of tracks,
- r_k: k-th resource,
- R: set of resources,
- π_k: path assigned to train v_k,
- ν_k: k-th vertex of a digraph,
- $D_\mu = (N_\mu, E_\mu)$: subdigraph of $D = (N, E)$,
- $D_R = (N_R, E_R)$: route digraph,
- $D_T(\boldsymbol{M}) = (N_R, E_T(\boldsymbol{M}))$: transition digraph at $\boldsymbol{M}$,
- $D_R^2 = (N^2, N_R^2)$: second level digraph,
- $\Gamma_u = (N_{\Gamma_u}, E_{\Gamma_u})$: strong subdigraph of D_R.

References

[1] Z.A. Banaszak, B.H. Krogh, "Dadlock Avoidance in Flexible Manufacturing Systems with Concurently Competing Process Flows," *IEEE Trans. on Robotics and Automation*, Vol. 6, No. 6, pp. 724-734, 1990.

[2] I. Britton, Links galore – railroad/railway links. http://www.britton2000.com/links/railroad.htm, 2000.

[3] D. Bromage, Railpage: Information on Australian Railways. http://www.railpage.org.au, 2002.

[4] Commission of the European Communities, White Paper on "European transport policy for 2010: time to decide," *COM (2001) 370*, 12/09/2001.

[5] B. De Schutter, T. van den Boom, "Model predictive control for railway networks," *2001 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, Como, Italy, July 2001.

[6] J. Ezpeleta, J.M. Colom, J. Martinez, "A Petri net based deadlock prevention policy for flexible manufacturing systems", *IEEE Trans. on Robotics and Automation*, Vol. 11, No. 2, pp. 173-184, 1995.

[7] M.P. Fanti, B. Maione, S. Mascolo, B. Turchiano, "Event based feedback control for deadlock avoidance in flexible production systems", *IEEE Trans. on Robotics and Automation*, Vol. 13, No. 3, pp. 347-363, 1997.

[8] M.P. Fanti, B. Maione, B. Turchiano, "Deadlock avoidance in flexible production systems with multiple capacity resources", *Studies in Informatics and Control*, Vol. 7, No. 4, pp. 343-364, 1998.

[9] M.P. Fanti, A. Giua, C. Seatzu, "A deadlock prevention method for railway networks using monitors for colored Petri nets," *2003 IEEE Int. Conf. on Systems, Man and Cybernetics*, Washington, USA, pp. 1866–1873, October 2003.

[10] M.P. Fanti, A. Giua, C. Seatzu, "Generalized mutual exclusion constraints and monitors for colored Petri nets," *2003 IEEE Int. Conf. on Systems, Man and Cybernetics*, Washington, USA, pp. 1860–1865, October 2003.

[11] A. Giua, F. DiCesare, M. Silva, "Generalized mutual exclusion constraints for nets with uncontrollable transitions," *IEEE Int. Conf. on Systems, Man and Cybernetics*, Chicago, USA, pp. 974–979, October 1992.

[12] A. Giua, C. Seatzu, "Liveness enforcing supervisors for railway networks using ES^{2}PR Petri nets," *Int. Workshop on Discrete Event Systems*, *6th Int. Workshop on Discrete Event Systems*, Zaragoza, Spain, pp. 361–366, October 2002.

[13] A. Giua, C. Seatzu, "Monitor design for Colored Petri nets with uncontrollable and unobservable transitions," *Int. Workshop on Discrete Event Systems*, *7th Int. Workshop on Discrete Event Systems*, Reims, France, September 2004.

[14] F. Harary, R.Z. Norman, D. Cartwright, *Structural models: an introduction to the theory of directed graphs*, John Wiley & Sons, Inc. New York, 1965.

[15] L.E. Holloway, B.H. Krogh, A. Giua, "A survey of Petri net methods for controlled discrete event systems", *Discrete Event Dynamic Systems: Theory and Application*, Vol. 7, No. 2, pp. 151–190, 1997.

[16] P.G. Howlett and P.J. Pudney, *Energy efficient train control*, Advances in Industrial Control, Springer Verlag, 1995.

[17] M. V. Iordache, J. O. Moody and P. J. Antsaklis, "Synthesis of deadlock prevention supervisors using Petri nets", *IEEE Trans. on Robotics and Automation*, Vol. 18, No. 1, pp. 59–68, February 2002.

[18] C.W. Janczura, "Modelling and analysis of railway network control logic using coloured Petri nets," *Ph.D. Thesis*, University of South Australia, August 1998.

[19] K. Jensen, *Coloured Petri Nets. Basic concepts, analysis methods and practical use. Volume 1: Basic Concepts*, EATCS Monographs on Theoretical Computer Science, Springer Verlag, 1992.

[20] K. Jensen, *Coloured Petri Nets. Basic concepts, analysis methods and practical use. Volume 2: Analysis methods*, EATCS Monographs on Theoretical Computer Science, Springer Verlag, 1994.

[21] M. Missikoff, "An object-oriented approach to an information and decision support systems for railway traffic control," *Emerging Application of Artificial Intelligence*, Vol. 11, pp. 25-40, 1998.

[22] J.O. Moody, P.J. Antsaklis, *Supervisory control of discrete event systems using Petri nets*, Kluwer Academic Publishers, 1998.

[23] T. Murata, "Petri Nets: properties, analysis and applications," *Proceedings of the IEEE*, Vol. 77, No. 4, pp. 541–580, April 1989.

[24] T.K.S. Murthy, L.S. Laurence, and R.E. Rivier, editors, *Computers in Railways Management*. Springer Verlag, 1987.

[25] T.K.S. Murthy, F.E. Young, S. Lehmann, and W.R. Smith, editors, *Computers in Railways Installations, Track and Signalling*. Springer Verlag, 1987.

[26] O.S. Nock, *Hystoric Railway Disasters*. Ian Allan, 1967.

[27] V.A. Profillidis, *Railway Engineering*. Avebury Technical, 1995.

[28] S.A. Reveliotis, M.A. Lawley, and P.M. Ferreira, "Polynomial-Complexity Deadlock Avoidance Policies for Sequential Resource Allocation Systems," *IEEE Trans. on Automatic Control*, Vol. 42, pp. 1344-1357, 1997.

[29] E.M. Reingold, J. Nievergelt, N. Deo, *Combinatorial algorithms. Theory and practice*, Prentice-Hall, Inc., New Jersey, 1977.

[30] N. Rezg, X.L. Xie, A. Ghaffari, "Supervisory control in discrete event systems using the theory of regions," *5th Workshop on Discrete Event Systems*, Ghent, Belgium, pp. 391–398, August 2000.

[31] G. Stremersch, R.K. Boel, "Structuring acyclic Petri nets for reachability analisys and control," *Discrete Event Dynamic Systems: Theory and Applications*, Vol. 12, No. 1, pp. 7–42, January 2002.

[32] E.A.G. Weits, "Simulation of railway traffic control," *Int. Trans. Operational Research*, Vol. 5, No. 6, pp. 461-469, 1998.