Optimization of Deterministic Timed Weighted Marked Graphs

Zhou He, Zhiwu Li, and Alessandro Giua

Abstract

Timed marked graphs, a special class of Petri nets, are extensively used to model and analyze cyclic manufacturing systems. Weighted marked graphs are convenient to model systems with bulk services and arrivals. We consider two problems of practical importance for this class of nets. The marking optimization problem consists in finding an initial marking to minimize the weighted sum of tokens in places while the average cycle time is less than or equal to a given value. The cycle time optimization problem consists in finding an initial marking to minimize the average cycle time while the weighted sum of tokens in places is less than or equal to a given value. We propose two heuristic algorithms to solve these problems. Several simulation studies show that the effectiveness of the proposed approach is significantly faster than existing ones.

Note to Practitioners—This paper is motivated by the optimization of manufacturing systems such as assembly lines and job-shops. The quantity of products which have to be stored or moved and the number and type of machines which operate the system have economical consequences. Therefore, the main problem for designers is to find a trade-off between minimizing the cost of the resources and maximizing the system's throughput. To this end, we formalize and address two problems. First, we aim to minimize the use of resources required to meet a desired throughput. Second, we aim to maximize the throughput with a given set of available resources. However, the existing results fail to provide practically effective and computationally efficient methods to analyze and solve the problems in such systems. The proposed iterative heuristic algorithms can find a proper schedule for a system, which is significant and useful to manufacturing engineers.

Published as:

Zhou He, Zhiwu Li, and A. Giua. "Optimization of Deterministic Timed Weighted Marked Graphs," *IEEE Trans. on Automation Science and Engineering,* Vol. 14, No. 2, pp. 1084-1095, 2017

DOI: 10.1109/TASE.2015.2490538

Zhou He is with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China. e-mail: zhouhe@stu.xidian.edu.cn

Zhiwu Li (corresponding author) is with the Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau, and also with School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China. e-mail: zhwli@xidian.edu.cn

Alessandro Giua is with Aix-Marseille Université, CNRS, ENSAM, Université de Toulon, LSIS UMR 7296, Marseille 13397, France and also with DIEE, University of Cagliari, Cagliari 09124, Italy. email: alessandro.giua@lsis.org, giua@diee.unica.it

I. INTRODUCTION

Timed Petri nets are a well known tool to model discrete event systems and represent their dynamic behavior. In this paper, we study a particular class of timed Petri nets called *timed weighted marked graphs* (TWMGs). The main structural feature of this class of nets is that each place has only one input and one output transition. Moreover, the firing delay of each transition is *deterministic*.

Timed weighted marked graphs and timed marked graphs (TMGs) find wide applications in manufacturing. They can model complex assembly lines and solve cyclic scheduling problems. Workshop operations and products are usually modeled by transitions and tokens, respectively. Between two successive transformations, semi-finished products have to be stored or moved from a workshop to another. The amount of products, also called Work In Process (WIP), that have to be stored or moved may have economical consequence. Therefore, the main problem for designers is to determine a proper schedule of WIP that allows the system to reach prespecified productivity while the amount of WIP is the smallest.

Some results have been developed for this class of Petri nets. Teruel *et al*. [1] proposed several techniques for the analysis of WMGs. Nakamura and Silva [2] and Campos *et al*. [3] developed methods to compute the average cycle time of TMGs for a given initial marking. Benabid-Najjar *et al.* [4] discussed the periodic schedules of bounded TWMGs and presented polynomial algorithms to check the existence of periodic schedules. Giua *et al*. [5] dealt with the problem of allocating a given number of tokens in a TMG so as to maximize the firing rate (i.e., the inverse of the cycle time) and proposed three different procedures to solve the problem. Laftit *et al*. [6] investigated the problem of reaching on TMGs a desired average cycle time that is less than a given value, while minimizing an invariant linear criterion. In their work, a heuristic algorithm and an exact algorithm are presented. However, in the literature, few studies are found to consider the marking optimization problem of TWMGs. Sauer [7] addressed the problem of finding an initial marking to minimize the weighted sum of tokens in places while the average cycle time is less than or equal to a given value, and proposed a heuristic solution based on an iterative process. Touris and Sauer [8] presented an approach based on the branch and bound to solve the same problem.

This paper copes with the marking optimization problem of a TWMG and the related cycle time optimization problem. The marking optimization problem consists in finding an initial marking to minimize the weighted sum of tokens in places while the average cycle time is less than or equal to a given value. The cycle time optimization problem consists in finding an initial marking to minimize the average cycle time while the weighted sum of tokens in places is less than or equal to a given value.

In the first part, we propose an iterative heuristic algorithm (Algorithm 1) to solve the marking optimization problem. At each step, we select places from some circuits to which we add tokens until the average cycle time is less than or equal to the desired value. Numerical simulation studies presented in Section IV-D show that the proposed method requires less iteration steps and thus is much faster than the approach in [7]. In some special cases the objective function that we obtained may be greater (i.e., worse) than the one found by Sauer. However, we show that by combining the two approaches, i.e., the one in [7] and the one proposed in this work, we can always reach the same objective function in [7] with a significant reduction of computational costs. Some of these results were presented in a preliminary form in [9].

In the second part of this paper, we study the cycle time optimization problem of a TWMG, an issue that to the best of our knowledge has not been addressed in the literature. Another iterative heuristic algorithm (Algorithm 2) is presented. At each step, we provide a criterion to select one place to which tokens are allocated until all available tokens (i.e., resources) are used up or the average cycle time reaches the lower bound.

The main contributions of the present paper can be summarized as follows:

1) A heuristic solution combined with an analytical method for the marking optimization problem is presented. By several experimental studies, we show that the combined method (He+Sauer) is more efficient than previous one in most cases.

2) The cycle time optimization problem of TWMGs is originally presented. A novel heuristic solution is proposed which is fast and efficient.

This paper is structured as follows. In the following section, we briefly recall some basic concepts and the main properties of TWMGs. Section III presents the problem statement. In Section IV, we propose a heuristic solution for the marking optimization problem based on a live marking. Following the algorithm, some numerical examples are shown to illustrate it and details are given to compare the proposed method with the one in [7]. In Section V, the cycle time optimization problem is considered and another heuristic algorithm is proposed. Some applications in manufacturing are presented at the end of this section. Conclusions are finally drawn in Section VI.

II. Background

A. Generalities

We assume that the reader is familiar with the structure, firing rules, and basic properties of Petri nets (see [1], [10] and [11]). In this section, we will recall the formalism used in the paper. A *place/transition* net (P/T net) is a structure $N = (P, T, Pre, Post)$, where P is a set of n *places*; T is a set of m *transitions*; $Pre : P \times T \rightarrow \mathbb{N}$ and $Post : P \times T \rightarrow \mathbb{N}$ are the *pre-* and *post-incidence* functions, respectively, that specify the arcs; and $C = Post - Pre$ is the *incidence matrix*. Note that $\mathbb{N}$ is a set of non-negative integers.

A vector $\boldsymbol{x} = [x_1, x_2, \ldots, x_m]^T \in \mathbb{N}^{|T|}$ such that $\boldsymbol{x} \neq \boldsymbol{0}$ and $C \cdot \boldsymbol{x} = \boldsymbol{0}$ is a *T-semiflow*. A vector $\boldsymbol{y} = [y_1, y_2, \ldots, y_n]^T \in \mathbb{N}^{|P|}$ such that $\boldsymbol{y} \neq \boldsymbol{0}$ and $\boldsymbol{y}^T \cdot C = \boldsymbol{0}^T$ is a *P-semiflow*. The *supports* of a T-semiflow and a P-semiflow are defined by $\|\boldsymbol{x}\|$=$\{t_i \in T | x_i > 0\}$ and $\|\boldsymbol{y}\|$=$\{p_i \in P | y_i > 0\}$, respectively. A *minimal* T-semiflow[1] (P-semiflow) is a T-semiflow $\|\boldsymbol{x}\|$ (P-semiflow $\|\boldsymbol{y}\|$) that is not a superset of the support of any other T-semiflow (P-semiflow), and its components are mutually prime.

A *marking* is a vector $\boldsymbol{M} : P \rightarrow \mathbb{N}$ that assigns to each place of a P/T net a non-negative integer number of tokens; we denote the marking of place p as $\boldsymbol{M}(p)$. A P/T *system* or *net system* $\langle N, \boldsymbol{M_0} \rangle$ is a net N with an

[1] This is also called a minimal and minimal support semiflow in some references. For the sake of simplicity, we call it a minimal semiflow.

initial marking $\boldsymbol{M_0}$. A transition t is enabled at $\boldsymbol{M}$ if $\boldsymbol{M} \geq Pre(\cdot, t)$ and an enabled transition t may fire yielding a marking $\boldsymbol{M'}$ with

$$\boldsymbol{M'} = \boldsymbol{M} + C(\cdot, t) \tag{1}$$

where $Pre(\cdot, t)$ (resp. $C(\cdot, t)$) denotes the column of the matrix Pre (resp. C) associated with transition t.

A P/T net is said to be *ordinary* when all arc weights are $1's$. A *marked graph* (MG, also called an event graph) is an ordinary Petri net that satisfies the condition $|^{\bullet}p| = |p^{\bullet}| = 1$. A *weighted marked graph* (WMG, also called a weighted event graph) is a net that also satisfies this condition but may not be ordinary, i.e., the weight associated with an arc can be a non-negative integer.

A deterministic timed P/T net is a pair $N^{\delta} = (N, \delta)$, where $N = (P, T, Pre, Post)$ is a P/T net, and $\delta : T \rightarrow \mathbb{N}$, called *firing delay*, assigns a non-negative integer firing duration to each transition. A transition with a firing delay equal to 0 is said to be immediate. A *clock* $\theta_i(t)$ associated with transition t represents the residual time to fire t at marking $\boldsymbol{M}_i$. If a transition t is not enabled at marking $\boldsymbol{M}_i$, it is usually assumed that $\theta_i(t) = +\infty$. We denote the *clock vector* $\boldsymbol{\theta_i} = (\theta_i(t_1), \theta_i(t_2), \ldots, \theta_i(t_m))$.

We denote the *state* of a TWMG as $[\boldsymbol{M}; \boldsymbol{\theta}]$. Thus the evolution of a deterministic TWMG will be described by the marking $\boldsymbol{M}_i$, the clock vector $\boldsymbol{\theta_i}$ and the time instant τ_i at which marking $\boldsymbol{M}_i$ is reached (for $i = 1, 2, \ldots, n$). We initialize $\tau_0 = 0$ for the initial marking $\boldsymbol{M}_0$.

When a transition t becomes enabled, it cannot fire before the time $\delta(t)$ has elapsed. Under the *As Soon As Possible* (ASAP) execution policy, a transition t will fire *exactly* after t is enabled for a time $\delta(t)$. In this paper, we deal with TWMGs which are conflict-free nets, and thus the firing of a transition t does not disable any other transitions. In the ASAP execution, from state $[\boldsymbol{M}_i; \boldsymbol{\theta_i}]$, all transitions that have a minimal value of the clock can be fired simultaneously after a time $\phi_i = \min_{t \in T} \theta_i(t) = \tau_{i+1} - \tau_i$ and the TWMG reaches a new state $[\boldsymbol{M}_{i+1}; \boldsymbol{\theta_{i+1}}]$. The marking will change as shown in Eq. (1) and the clock will be updated by the following equations:

$$\theta_0(t) = \begin{cases} \delta(t), & if\ \boldsymbol{M}_0[t\rangle \\ +\infty, & if\ \neg \boldsymbol{M}_0[t\rangle \end{cases} \tag{2}$$

$$\theta_i(t) = \begin{cases} \delta(t), & if\ \neg \boldsymbol{M}_{i-1}[t\rangle\ and\ \ \boldsymbol{M}_i[t\rangle \\ & or\ t\ fired\ at\ \boldsymbol{M}_{i-1}\ and\ \boldsymbol{M}_i[t\rangle \\ +\infty, & if\ \neg \boldsymbol{M}_i[t\rangle \\ \theta_{i-1}(t) - \phi_{i-1}, & otherwise \end{cases} \tag{3}$$

where $\neg \boldsymbol{M}[t\rangle$ means that t is not enabled at marking $\boldsymbol{M}$. In this paper, we consider a single server semantics, i.e., we assume that each transition can fire only once at each time instant even if its enabling degree is greater than one (see [7]).

B. Strongly connected nets and circuits

A net is *strongly connected* if there exists a directed path from any node in $P \cup T$ to every other node. Let us define an *elementary circuit* γ (or elementary cycle) in a net as a directed path that goes from one node back to the same node without passing twice on the same node. In a strongly connected net, it is easy to show that each node belongs to an elementary circuit, and thus the name *cyclic nets* are also used to denote this class.

Given a place p of a WMG, let t_i (resp. t_j) be its unique input (resp. output) transition as shown in Fig. 1. We denote by $w(p) = Post(p, t_i)$ the weight of its input arc and by $\nu(p) = Pre(p, t_j)$ the weight of its output arc. For any place $p \in P$, we denote by gcd_p (resp. lcm_p) the *greatest common divisor* (resp. the *least common multiple*) of the integers $w(p)$ and $\nu(p)$.

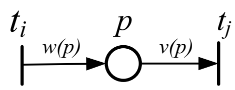


Fig. 1. A place p between two transitions t_i and t_j.

Definition 1: Every elementary circuit γ of a WMG is neutral if the following condition holds.

$$\prod_{p \in \gamma} \frac{\nu(p)}{w(p)} = 1$$

■

In other words, in a neutral circuit the product of the weights of all pre-arcs is equal to that of the weights of all post-arcs. This means that if the circuit initially contains enough tokens, it is possible to fire all transitions along the path returning to the same initial marking. It is well known that a WMG whose circuits are all neutral has a unique T-semiflow $\boldsymbol{x}$ and it contains all transitions in its support [1]. In this paper, we limit our study to strongly connected WMGs in which all circuits are neutral.

A strongly connected WMG in which all circuits are neutral is *bounded*, i.e., there exists an integer $B \in \mathbb{N}$ such that the marking of any place p is not greater than B at any reachable marking.

C. Liveness of a TWMG

Theorem 1: (Theorem 4.12 of Teruel *et al.* [1]) *A TWMG is live iff each elementary circuit is live.* ■

In the case of a TMG, an elementary circuit is live if there exists at least one token in the circuit. The liveness decision problem of a TMG is polynomial, solved in [12], [13]. A weighted circuit of a TWMG is live if each transition can be fired infinitely. However, determining the liveness of a weighted circuit is not so easy. Up to now, no polynomial algorithm for liveness checking has been found, for example, the algorithms developed in [14] to answer this question are not polynomial. Next, we review some sufficient conditions for the liveness of weighted circuits existing in the literature. Later, these conditions will be used in the proposed optimization approach.

Teruel *et al.* [1] and Chrzastowski-Wachtel and Raczunas [15] proposed a few methods to verify the liveness of weighted circuits. First they define a weighted function with respect to a marking, i.e.,

$$W(\boldsymbol{M}) = \boldsymbol{y}^T \cdot \boldsymbol{M} \tag{4}$$

where $\boldsymbol{y}$ is a minimal P-semiflow. Furthermore, they define a marking:

$$\boldsymbol{M}_D = (\nu(p_1) - 1, \nu(p_2) - 1, \ldots, \nu(p_n) - 1)^T \tag{5}$$

The following result provides a sufficient, albeit restrictive, condition for liveness.

Proposition 1: (Proposition 5.2 of Teruel *et al.* [1]) *If $W(\boldsymbol{M}_0) > W(\boldsymbol{M}_D)$, then the weighted circuit is live.* ■

Less restrictive conditions for liveness also exist. Let $\mathbb{R}^+$ be a set of positive real numbers and $\mathcal{M}(\omega) = \{\boldsymbol{M} | W(\boldsymbol{M}) = \omega, \omega \in \mathbb{R}^+\}$. The *least live weight* is the minimal ω such that $\forall \boldsymbol{M} \in \mathcal{M}(\omega)$, $\boldsymbol{M}$ is a live marking. In [15] the least live weight of a weighted circuit with a minimal P-semiflow $\boldsymbol{y}$ was defined as

$$W_L = W(\boldsymbol{M}_D) - g(y_1, y_2, \cdots, y_n) \tag{6}$$

where g is the Frobenius number.[2] Note that a Frobenius number only exists if all its arguments are greater than one and coprime. The first condition is always verified in our case since we consider minimal P-semiflows. The second condition may not always be verified: when it is, the least live weight in Eq. (6) can be computed and the following proposition holds.

Proposition 2: (Proposition 2.4 of Chrzastowski-Wachtel and Raczunas [15]) *If $g(y_1, y_2, \cdots, y_n)$ has no non-negative integer solution and the marking $\boldsymbol{M}_0$ satisfies $W(\boldsymbol{M}_0) = W_L$, then the weighted circuit is live.* ■

In the case that there exists a unitary component in a minimal P-semiflow, then a least live weight cannot be computed by Eq. (6). Let us consider the example in Fig. 2. We have $\boldsymbol{y}^T$ = (3, 4, 3), $\boldsymbol{M}_D$ = (3, 2, 2)^T, $\boldsymbol{x}$ = (4, 3, 3)^T, and W_L = $W(\boldsymbol{M}_D) - g(y_1, y_2, y_3)$ = 23$-g$(3, 4, 3) =23$-$5=18. We conclude that any marking $\boldsymbol{M}$ with weight $W(\boldsymbol{M}) > 23$ or $W(\boldsymbol{M}) = 18$ is a live marking.

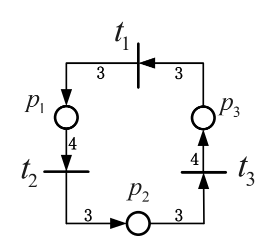


Fig. 2. A weighted circuit.

[2]Given positive integers y_1, y_2, $\cdots$, y_n such that gcd(y_1, y_2, $\cdots$, y_n)=1, the Frobenius number g(y_1, y_2, $\cdots$, y_n) is the largest integer that cannot be expressed as an integer linear combination of these numbers, i.e., as a sum $a_1y_1 + a_2y_2 + \cdots + a_ny_n$, where a_1, a_2, $\cdots$, and a_n are non-negative integers.

It can be checked that every marking with a weight equal to 18 is live. For instance (6, 0, 0)^T as well as (0, 3, 2)^T is live. We use the two approaches above to select a live initial marking.

D. Cycle time of a TWMG

The *average cycle time* $\chi(\boldsymbol{M})$ of a TWMG system $\langle N, \boldsymbol{M}\rangle$ is the average time to fire once the minimal T-semiflow under the ASAP operational model (i.e., transitions are fired as soon as possible). For deterministic TWMGs, the following limit exists:

$$\lim_{\tau\to\infty} \frac{\vec{\sigma}_\tau}{\tau} = \vec{\sigma}^* < \vec{\infty}$$

where the vector $\vec{\sigma}_\tau$ represents the firing vector from time 0 to time τ and the constant vector $\vec{\sigma}^*$ is called the limit firing vector. $\vec{\sigma}^*(t_i)$ represents the average number of firing t_i per time unit. The average cycle time of transition t_i of a TWMG is the average time between two consecutive firings of t_i, which is equal to

$$\frac{1}{\vec{\sigma}^*(t_i)}$$

Definition 2: Let $t_i \in T$ be an arbitrary transition of a TWMG with the minimal T-semiflow $\boldsymbol{x}$. The average cycle time *of the TWMG is equal to*

$$\frac{x_i}{\vec{\sigma}^*(t_i)}$$

■

Note that the value of the average cycle time does not depend on the considered transition. In [16] and [17], the authors proved that the ASAP execution of a live and strongly connected TMG with integer delays is ultimately repetitive. The period of the pattern is $\tau_{ij} = \tau_j - \tau_i$ and the number of firings of every transition within a period is k (the periodicity). In terms of TWMGs, the ASAP execution is also ultimately periodic. Fig. 3 shows the evolution of a live TWMG, where $\boldsymbol{M}_0$ is the initial marking and the arrows correspond to ASAP execution steps. The number of firings of transition t_i within the steady period is k_i. This value is not identical for each transition but the proportion is equal to the minimal T-semiflow. Thus, the average cycle time of the TWMG is equal to

$$x_i \cdot \frac{\tau_{ij}}{k_i} \tag{7}$$

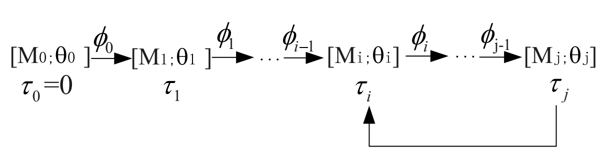


Fig. 3. Evolution of a live and strongly connected TWMG.

The average cycle time of a TWMG depends on the average cycle time of its circuits. Considering a net consisting of only one circuit, we define $\chi_\gamma(\boldsymbol{M})$ as the average cycle time of circuit γ. Let Γ represent the set of elementary circuits of a cyclic TWMG and $\chi^\star(\boldsymbol{M}) = \max_{\gamma\in\Gamma} \chi_\gamma(\boldsymbol{M})$ be a critical time. Any $\gamma \in \Gamma$ such that $\chi_\gamma(\boldsymbol{M}) =$

$\chi^\star(\boldsymbol{M})$ is a critical circuit that is denoted as $\gamma^\star$. It is well known that for a TMG the average cycle time is equal to the critical time, i.e.,

$$\chi(\boldsymbol{M}) = \chi^\star(\boldsymbol{M})$$

This result does not apply to a TWMG as we will show in the example in Fig. 5, but it holds that the critical time is less than or equal to the average cycle time, i.e.,

$$\chi(\boldsymbol{M}) \geq \chi^\star(\boldsymbol{M})$$

To a certain extent, the critical circuits are those that bind the speed of the system. Up to now, no analytical method to compute the average cycle time of a TWMG has been proposed in the literature: only bounds on this value can be found in [3]. The studies in [2], [14] have proposed techniques to convert a TWMG into an equivalent TMG for which an average cycle time can be analytically computed. However, this expansion can lead to a model of significant size. A method to compute the average cycle time of a TWMG is proposed in [19] but under restrictive conditions on the initial marking. This is why in this paper we resort to a simulation study to compute the average cycle time of a TWMG using Eq. (7).

Example: Let us consider the evolution of the TWMG in Fig. 4(b). It starts with an initial marking $\boldsymbol{M}_0 = (10, 0)^T$ and $\boldsymbol{\theta}_0 = (+\infty, 5)$. The evolution of the TWMG is presented as follows:

$$[M_0=(10,0)^{\mathrm{T}};\theta_0=(+\infty,5);\tau_0=0]$$
$$\mathrm{t}_2\downarrow$$
$$[M_1=(6,4)^{\mathrm{T}};\theta_1=(+\infty,5);\tau_1=5]$$
$$\mathrm{t}_2\downarrow$$
$$[M_2=(2,8)^{\mathrm{T}};\theta_2=(2,+\infty);\tau_2=10]$$
$$\mathrm{t}_1\downarrow$$
$$[M_3=(8,2)^{\mathrm{T}};\theta_3=(+\infty,5);\tau_3=12]$$
$$\mathrm{t}_2\downarrow$$
$$[M_4=(4,6)^{\mathrm{T}};\ \theta_4=(2,5);\ \tau_4=17]$$
$$\mathrm{t}_1\downarrow$$
$$[M_5=(10,0)^{\mathrm{T}};\theta_5=(+\infty,3);\tau_5=19]$$
$$\mathrm{t}_2\downarrow$$
$$[M_6=(6,4)^{\mathrm{T}};\theta_6=(+\infty,5);\tau_6=22]$$

Note that states $[\boldsymbol{M}_1; \boldsymbol{\theta_1}]$ and $[\boldsymbol{M}_6; \boldsymbol{\theta_6}]$ are the same, implying that from state $[\boldsymbol{M}_6; \boldsymbol{\theta_6}]$ the system will enter a cycle which includes five states and the firing sequence is $t_2t_1t_2t_1t_2$. The minimal T-semiflow is $\boldsymbol{x} = (2, 3)^T$ and $\tau_{1,6} = 17$. The average cycle time of the TWMG is equal to 17 by solving Eq. (7).

III. Problem statement

A. Problem formulation

The *marking optimization problem* of a TWMG consists in finding an initial marking $\boldsymbol{M}_0$ that minimizes a weighted function of the initial marking while the average cycle time is less than or equal to a given value. In other words we aim to find a solution of the following optimization problem:

$$(P1)\begin{cases} min\ f(\boldsymbol{M}_0) = \boldsymbol{y}^T \cdot \boldsymbol{M}_0 \\ s.t.\ \chi(\boldsymbol{M}_0) \leq b \end{cases} \tag{8}$$

where

- $\chi(\boldsymbol{M}_0)$ is the average cycle time of the TWMG associated with marking $\boldsymbol{M}_0$,
- b is a given positive real number that represents the upper bound of the cycle time, and
- $\boldsymbol{y} = \sum_{\gamma\in\Gamma} c_\gamma \cdot \boldsymbol{y}_\gamma$ is a P-semiflow and $\boldsymbol{y}_\gamma$ is a minimal P-semiflow, i.e.,

$$y_\gamma(p_i) = y(p_i)\ if\ p_i \in \gamma,\ else\ y_\gamma(p_i) = 0$$

and c_γ represents the cost of the resources modeled by tokens in the support of $\boldsymbol{y}_\gamma$.

We choose $\boldsymbol{y}$ as a P-semiflow since the value of $\boldsymbol{y}^T \cdot \boldsymbol{M}_0$ at every reachable marking $\boldsymbol{M} \in R(N, \boldsymbol{M}_0)$ is invariant.

Proposition 3: (Proposition 2 of Sauer [7]) *$(P1)$ has a solution iff* $b \geq \chi'(\boldsymbol{M}) = max\{x_i \cdot \delta_i, t_i \in T\}$, where $\boldsymbol{x}$ is the minimal T-semiflow. ■

The *cycle time optimization problem* of a TWMG consists in finding an initial marking that minimizes the average cycle time, i.e., maximizes the throughput of the system, while the weighted sum of the marking (resources) is less than or equal to a given value. In other words we need to find a solution of the following optimization problem:

$$(P2)\begin{cases} min\ \chi(\boldsymbol{M_0}) \\ s.t.\ \boldsymbol{y}^T \cdot \boldsymbol{M}_0 \leq s \end{cases} \tag{9}$$

where

- $\boldsymbol{y}$ is a P-semiflow as in $(P1)$,
- s is a given positive real number, representing the maximal available resources that can be allocated.

Proposition 4: $(P2)$ has a finite solution if $s \geq s^\star$, where

$$s^\star = min\ \ \boldsymbol{y}^T \cdot \boldsymbol{M}$$

$$s.t.\ \boldsymbol{y}_\gamma^T \cdot \boldsymbol{M} > W(\boldsymbol{M}_D^\gamma), \forall\gamma \in \Gamma \qquad ■$$

Proof. Let $\boldsymbol{M}$ be the initial marking of a TWMG and $W(\boldsymbol{M}_D^\gamma)$ be the weighted function of marking $\boldsymbol{M}_D$ restricted to circuit γ. If $\forall\gamma \in \Gamma$, $\boldsymbol{y}_\gamma^T \cdot \boldsymbol{M} > W(\boldsymbol{M}_D^\gamma)$, from Proposition 1, we conclude that each circuit of the TWMG is live. Then, the TWMG is necessarily live according to Theorem 1 and its average cycle time will be finite. □

B. A previous approach for Problem $P1$

In this subsection, we briefly recall an approach dealing with the marking optimization problem of TWMGs presented by Sauer in [7].

The proposed iterative heuristic algorithm starts with an initial marking $\boldsymbol{M}_0$ such that

$$\boldsymbol{M}_0(p) = x_{p^\bullet} \cdot Pre(p, p^\bullet),\ \forall p \in P.$$

Obviously, under the condition imposed by Proposition 3 this marking is feasible for Eq. (8), i.e., it satisfies $\chi(\boldsymbol{M}_0) \leq b$.

The approach requires to evaluate the average cycle time and the corresponding average marking by simulation. At each iteration step, one place is selected to remove a token from $\boldsymbol{M}_0$ as long as the average cycle time is less than or equal to the upper bound b. The selected place should maximize the following criterion:

$$L(p, \propto) \cdot y_p,$$

where $L(p, \propto)$ denotes the number of tokens in the average marking that cannot be used to enable transition $p^\bullet$. If $\boldsymbol{M}_0(p) = 0$, a marking reachable from $\boldsymbol{M}_0$ containing at least one token is computed. The algorithm stops when there is no place that can be selected to remove tokens.

When the net size becomes larger, this approach usually requires a large number of iteration steps to remove the redundant tokens.

IV. Marking optimization for TWMGs

A. *The main idea of the heuristic solution*

We propose here a fast and efficient heuristic solution based on an iterative process to solve the problem of marking optimization for TWMGs. It starts with a live marking that has a *small* weighted sum, and then we compute the average cycle time of the TWMG. If the average cycle time is greater than the upper bound of the cycle time, we add tokens to some circuits until the average cycle time is less than or equal to the upper bound of the cycle time. We select the places to which tokens should be added so as to increase the performance index $f(\boldsymbol{M}_0)$ as small as possible.

1) Useful tokens: The initial marking $\boldsymbol{M}_0(p)$ of any place p can be replaced by $\boldsymbol{M}_0^\star(p)$ tokens without any influence on the precedence constraints induced by p (see [20] and [21]), where

$$\boldsymbol{M}_0^\star(p) = \left\lfloor \frac{\boldsymbol{M}_0(p)}{gcd_p} \right\rfloor \cdot gcd_p \tag{10}$$

As a result, we can deduce that the average cycle time at $\boldsymbol{M}_0$ and $\boldsymbol{M}_0^\star$ are the same. However, the value of $f(\boldsymbol{M}_0^\star)$ is less than or equal to $f(\boldsymbol{M}_0)$. Let us consider the example in Fig. (4). The initial marking of the TWMG is $\boldsymbol{M}_0 = (11, 1)^T$ and gcd_{p_1} = gcd_{p_2} = 2.

$$\boldsymbol{M}_0^\star(p_1) = \left\lfloor \frac{\boldsymbol{M}_0(p_1)}{gcd_{p_1}} \right\rfloor \cdot gcd_{p_1} = \left\lfloor \frac{11}{2} \right\rfloor \cdot 2 = 10$$
$$\boldsymbol{M}_0^\star(p_2) = \left\lfloor \frac{\boldsymbol{M}_0(p_2)}{gcd_{p_2}} \right\rfloor \cdot gcd_{p_2} = \left\lfloor \frac{1}{2} \right\rfloor \cdot 1 = 0$$

Then $\boldsymbol{M}_0^\star = (10, 0)^T$ and we can check that $f(\boldsymbol{M}_0^\star) = 10 < f(\boldsymbol{M}_0) = 12$ and the average cycle time at $\boldsymbol{M}_0$ and $\boldsymbol{M}_0^\star$ are identical, i.e., $\chi(\boldsymbol{M}_0)$ = $\chi(\boldsymbol{M}_0^\star) = 17$.

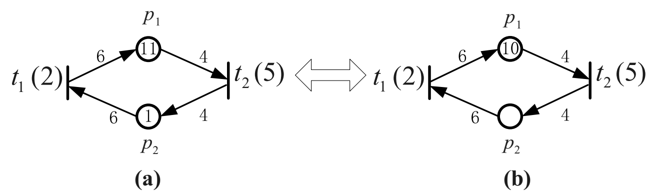


Fig. 4. Useful tokens.

2) Selection of a proper initial marking: For each circuit, there exist some markings that satisfy the least live weight condition. We choose the one that makes the net live while satisfies the following condition:

$$\begin{cases} min \; f(\boldsymbol{M}) = \boldsymbol{y}^T \cdot \boldsymbol{M} \\ s.t. \quad C(\boldsymbol{M}, \gamma) \quad \forall \gamma \in \Gamma \end{cases} \tag{11}$$

where

$$C(\boldsymbol{M}, \gamma) : \boldsymbol{y}_\gamma^T \cdot \boldsymbol{M} = W_L^\gamma \tag{12}$$

or

$$C(\boldsymbol{M}, \gamma) : \boldsymbol{y}_\gamma^T \cdot \boldsymbol{M} > W(\boldsymbol{M}_D^\gamma) \tag{13}$$

For each circuit γ, we consider its minimal P-semiflow. If it contains no unitary component, the least live weight W_L^γ of the circuit can be determined and we use Eq. (12) for γ, as this provides a sufficient condition for liveness with minimal cost. If the minimal P-semiflow of the circuit contains unitary components, we use Eq. (13).

We point out that it may happen that IPP (11) has no feasible solution due to the presence of the equality constraints given by Eq. (12) that may not be compatible. Should this situation occur, we use for all circuits the inequality constraints given by Eq. (13), thus ensuring that a feasible solution exists.

When there exists more than one optimal solution for the marking $\boldsymbol{M}$, we choose one. Then the initial marking $\boldsymbol{M}_0$ can be computed using Eq. (10), i.e., $\boldsymbol{M}_0 = \boldsymbol{M}^\star$. If we start the iteration from a marking that satisfies the condition above, we can ensure that the net is live and the value of performance index $f(\boldsymbol{M}_0)$ is small. If the average cycle time of $\boldsymbol{M}_0$ is greater than the upper bound of the cycle time, we add tokens to the net until the requirement on the average cycle time is satisfied. Otherwise, the initial marking $\boldsymbol{M}_0$ is a heuristically good solution (although possibly not optimal).

3) Selection of the places to add tokens: After we select the initial marking $\boldsymbol{M}_0$, we can compute the average cycle time $\chi(\boldsymbol{M}_0)$ of the TWMG and $\chi_\gamma(\boldsymbol{M}_0)$ for every elementary circuit. If the average cycle time satisfies the condition $\chi(\boldsymbol{M}_0) \leq b$, no more tokens should be added and the marking $\boldsymbol{M}_0$ is chosen as a solution.

If the average cycle time does not satisfy the condition $\chi(\boldsymbol{M}_0) \leq b$, two situations are possible. If there exist circuits γ's that have average cycle time greater than b, i.e., $\chi_\gamma(\boldsymbol{M}_0) > b$, tokens should be added to all these circuits. The set of selected circuits is denoted as $\Gamma_c = \{\gamma \in \Gamma | \chi_\gamma(\boldsymbol{M}_0) > b\}$. However, as we discussed in II-D,

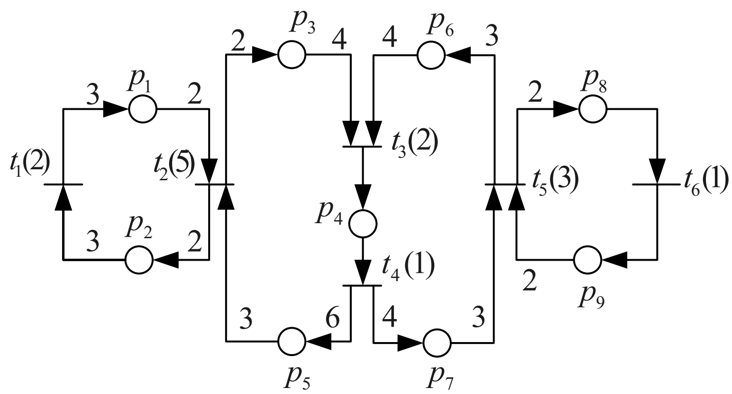


Fig. 5. The TWMG of Example 1.

it may also happen that for any circuit $\gamma \in \Gamma$, $\chi_\gamma(\boldsymbol{M}_0) \leq b$ holds, even if the cycle time of the net is $\chi(\boldsymbol{M}_0) > b$. In this case we choose to add tokens to all critical circuits $\gamma^\star$, i.e., the set of circuits selected for adding tokens is $\Gamma_c = \{\gamma \in \Gamma | \chi_\gamma(\boldsymbol{M}_0) = \chi^\star(\boldsymbol{M}_0)\}$, where $\chi^\star(\boldsymbol{M}_0)$ is the critical time.

For each of these circuits, we select one place p_r and add gcd_{p_r} tokens to this place. We choose the one that increases $f(\boldsymbol{M}_0)$ as small as possible, i.e., the increment of the criterion value $f(\boldsymbol{M}_0)$ should be the least after adding gcd_{p_r} tokens. We define an n-dimensional vector $\boldsymbol{I}$ of zeros and ones.

$$\boldsymbol{I}^T = (I_{p_1}, I_{p_2}, \cdots, I_{p_n}) \tag{14}$$

where

$$I_{p_r} = \begin{cases} 1, & add \;\; gcd_{p_r} \;\; tokens \;\; to \;\; place \;\; p_r \\ 0, & add \;\; 0 \;\; token \;\; to \;\; place \;\; p_r \end{cases} \tag{15}$$

In other words, we add tokens to the places with the coefficient I_{p_r} = 1. Let P_a be the set of these places and $\boldsymbol{G_d}$ = $(gcd_{p_1} \cdot y_1,\, gcd_{p_2} \cdot y_2,\, \cdots,\, gcd_{p_n} \cdot y_n)^T$, where $\boldsymbol{y}$ is a P-semiflow of the net and $gcd_{p_r} \cdot y_r$ represents the increment of $f(\boldsymbol{M}_0)$ after adding gcd_{p_r} tokens to place p_r. We denote by $\Delta f(\boldsymbol{M}_0)$ the total increment of $f(\boldsymbol{M}_0)$, where

$$\Delta f(\boldsymbol{M}_0) = \boldsymbol{I}^T \cdot G_d \tag{16}$$

Then, we can select the places by solving the following problem:

$$\begin{cases} \min \quad \Delta f(\boldsymbol{M}_0) \\ s.t. \;\; \sum_{p \in \gamma} I_p = 1, \;\; \forall \gamma \in \Gamma_c \end{cases} \tag{17}$$

The constrains in Eq. (17) will ensure that only one place should be selected for each circuit.

B. Heuristic solution

We can summarize the proposed procedure in Algorithm 1. In step 6 of Algorithm 1, the average cycle time needs to be computed. This can be done using any of the techniques mentioned in Section II-D. In this paper, we use the Petri net tool HYPENS [24] to compute the cycle time via simulation.

Algorithm 1: Marking Optimization

Input: A cyclic TWMG N with a set of elementary circuits Γ, an upper bound b of its average cycle time, and a P-semiflow $\boldsymbol{y} = \sum_{\gamma \in \Gamma} c_\gamma \cdot \boldsymbol{y}_\gamma$.

Output: An initial marking $\boldsymbol{M}_0$ such that the cycle time of the net satisfies $\chi(\boldsymbol{M}_0) \leq b$.

1: Compute the marking $\boldsymbol{M}_D$ as in Eq. (5).

2: For every elementary circuit $\gamma \in \Gamma$, compute $W(\boldsymbol{M}_D^\gamma) = \boldsymbol{y}_\gamma^T \cdot \boldsymbol{M}_D$.

3: For every elementary circuit $\gamma \in \Gamma$, compute $W_L^\gamma = \boldsymbol{y}_\gamma^T \cdot \boldsymbol{M}_D - g$ if possible.

4: Compute a marking $\boldsymbol{M}$ that satisfies Eq. (11).

5: Compute an initial marking $\boldsymbol{M}_0 = \boldsymbol{M}^\star$.

6: Compute the average cycle time $\chi(\boldsymbol{M}_0)$ and $\chi_\gamma(\boldsymbol{M}_0)$, $\forall \gamma \in \Gamma$.

7: If $\chi(\boldsymbol{M}_0) \leq b$, stop and $\boldsymbol{M_0}$ is a solution.

8: while ($\chi(\boldsymbol{M}_0) > b$)

{

If $\exists \gamma$, $\chi_\gamma(\boldsymbol{M}_0) > b$, tokens should be added to all these circuits in $\Gamma_c = \{\gamma \in \Gamma | \chi_\gamma(\boldsymbol{M}_0) > b\}$

Else $\Gamma_c = \{\gamma \in \Gamma | \chi_\gamma(\boldsymbol{M}_0) = \chi^\star(\boldsymbol{M}_0)\}$;

Compute I and P_a;

Add tokens to P_a and update $\boldsymbol{M}_0$;

}

9: Output an initial marking $\boldsymbol{M}_0$.

TABLE I

THE ITERATION PROCESS FOR EXAMPLE 1.

$\boldsymbol{M}_0$	$\chi_{\gamma_1}(\boldsymbol{M}_0)$	$\chi_{\gamma_2}(\boldsymbol{M}_0)$	$\chi_{\gamma_3}(\boldsymbol{M}_0)$	$\chi_{\gamma_4}(\boldsymbol{M}_0)$	$\chi(\boldsymbol{M}_0)$	b	$f(\boldsymbol{M}_0)$	Γ_c	P_a
$(4, 0, 4, 0, 0, 0, 6, 2, 0)^T$	38	39	21	20	43	30	34	$\{\gamma_1, \gamma_2\}$	$\{p_1, p_5\}$
$(5, 0, 4, 0, 3, 0, 6, 2, 0)^T$	34	30	21	20	34	30	42	$\{\gamma_1\}$	$\{p_1\}$
$(6, 0, 4, 0, 3, 0, 6, 2, 0)^T$	30	30	21	20	30	30	44		

C. Numerical Examples

We consider Example 1 in Fig. 5. There are four weighted circuits in the TWMG:

$$\begin{cases} \gamma_1 = p_1 t_2 p_2 t_1 \\ \gamma_2 = p_3 t_3 p_4 t_4 p_5 t_2 \\ \gamma_3 = p_6 t_3 p_4 t_4 p_7 t_5 \\ \gamma_4 = p_8 t_6 p_9 t_5 \end{cases}$$

The minimal T-semiflows of γ_1, γ_2, γ_3, and γ_4 are

$$\begin{cases} \boldsymbol{x}_1 = (2,3,0,0,0,0)^T \\ \boldsymbol{x}_2 = (0,2,1,1,0,0)^T \\ \boldsymbol{x}_3 = (0,0,3,3,4,0)^T \\ \boldsymbol{x}_4 = (0,0,0,0,1,2)^T \end{cases}$$

while the minimal P-semiflows of γ_1, γ_2, γ_3, and γ_4 are

$$\begin{cases} \boldsymbol{y}_1 = (1,1,0,0,0,0,0,0,0)^T \\ \boldsymbol{y}_2 = (0,0,3,12,2,0,0,0,0)^T \\ \boldsymbol{y}_3 = (0,0,0,4,0,1,1,0,0)^T \\ \boldsymbol{y}_4 = (0,0,0,0,0,0,0,1,1)^T \end{cases}$$

TABLE II

THE ITERATION PROCESS FOR EXAMPLE 2.

$\boldsymbol{M}_0$	$\chi_{\gamma_1}(\boldsymbol{M}_0)$	$\chi_{\gamma_2}(\boldsymbol{M}_0)$	$\chi(\boldsymbol{M}_0)$	b	$f(\boldsymbol{M}_0)$	Γ_c	P_a
$(2, 3, 0, 1, 0)^T$	29	26	30	21	20	$\{\gamma_1, \gamma_2\}$	$\{p_3\}$
$(2, 3, 2, 1, 0)^T$	25	26	26	21	24	$\{\gamma_1, \gamma_2\}$	$\{p_3\}$
$(2, 3, 4, 1, 0)^T$	21	21	21	21	28		

TABLE III

A COMPARISON BETWEEN THE APPROACH OF SAUER AND THE APPROACH PROPOSED IN THIS PAPER (HE).

				Sauer [ave]			He [ave]			He/Sauer [ave]		
Nb. of cycles	Nb. of nets	$\|P\|$ [ave]	$\|T\|$ [ave]	Iteration steps	CPU time [s]	Obj. Fun.	Iteration steps	CPU time [s]	Obj. Fun.	Iteration steps	CPU time [s]	Obj. Fun.
1	10	4	4	36.5	168	29.5	3.6	18	29.7	26.4%	29.2%	101.0%
2	10	9	8	64.7	615	34.3	1.9	44	38.5	9.4%	26.8%	111.3%
4	10	15	12	279.7	3676	80.2	3.6	155	85.8	2.2%	9.4%	106.7%
6	10	22	17	387.5	8890	100.8	4	358	114.1	1.5%	6.0%	114.1%
10	10	40	31	—	—	—	4.3	753	191.5	4.3/—	753/—	191.5/—

The cost of γ_1 and γ_3 is twice the cost of γ_2 and γ_4, i.e., $c_{\gamma_1} = c_{\gamma_3} = 2$ and $c_{\gamma_2} = c_{\gamma_4} = 1$. Therefore, the P-semiflow used in the criterion $f(\boldsymbol{M}_0)$ is $\boldsymbol{y} = 2\boldsymbol{y}_1 + \boldsymbol{y}_2 + 2\boldsymbol{y}_3 + \boldsymbol{y}_4$= (2, 2, 3, 20, 2, 2, 2, 1, 1)^T, and the minimal T-semiflow of the net is $\boldsymbol{x}$=(4, 6, 3, 3, 4, 8)^T.

$$\begin{cases} \gamma_1 : since\ y_1 = 1,\ W(\boldsymbol{M}_D) = 1 \times 1 + 1 \times 2 = 3 \\ \gamma_2 : W_L = W(\boldsymbol{M}_D) - g(y_3, y_4, y_5) = 13 - 1 = 12 \\ \gamma_3 : since\ y_6 = 1,\ W(\boldsymbol{M}_D) = 4 \times 0 + 1 \times 2 + 1 \times 3 = 5 \\ \gamma_4 : since\ y_8 = 1,\ W(\boldsymbol{M}_D) = 1 \times 0 + 1 \times 1 = 1 \end{cases}$$

We have gcd_{p_1} = 1, gcd_{p_2} = 1, gcd_{p_3} = 2, gcd_{p_4} = 1, gcd_{p_5} = 3, gcd_{p_6} = 1, gcd_{p_7} = 1, gcd_{p_8} = 1, and gcd_{p_9} = 1.

$$\boldsymbol{G_d} = (2, 2, 6, 20, 6, 2, 2, 1, 1)^T$$

$$\min \ \ f(\boldsymbol{M}) = 2\boldsymbol{M}(p_1) + 2\boldsymbol{M}(p_2) + 3\boldsymbol{M}(p_3) + 20\boldsymbol{M}(p_4) + 2\boldsymbol{M}(p_5) + 2\boldsymbol{M}(p_6) + 2\boldsymbol{M}(p_7) + \boldsymbol{M}(p_8) + \boldsymbol{M}(p_9)$$

$$s.t \begin{cases} \boldsymbol{M}(p_1) + \boldsymbol{M}(p_2) > 3 \\ 3\boldsymbol{M}(p_3) + 12\boldsymbol{M}(p_4) + 2\boldsymbol{M}(p_5) = 12 \\ 4\boldsymbol{M}(p_4) + \boldsymbol{M}(p_6) + \boldsymbol{M}(p_7) > 5 \\ \boldsymbol{M}(p_8) + \boldsymbol{M}(p_9) > 1 \end{cases}$$

We obtain a marking $\boldsymbol{M} = (4, 0, 4, 0, 0, 0, 6, 2, 0)^T$ and the initial marking $\boldsymbol{M}_0 = \boldsymbol{M}^\star = \boldsymbol{M}$. From Table I, we can find that the average cycle time of γ_1 and γ_2 are greater than the upper bound of the cycle time b at the initial marking $\boldsymbol{M}_0$. Then, we compute $\boldsymbol{I}^T$ and P_a to add tokens.

$$\min \ \ \Delta f(\boldsymbol{M}_0) = 2I_{p_1} + 2I_{p_2} + 6I_{p_3} + 20I_{p_4} + 6I_{p_5} + 2I_{p_6} + 2I_{p_7} + I_{p_8} + I_{p_9}$$

$$s.t \begin{cases} I_{p_1} + I_{p_2} = 1 \\ I_{p_3} + I_{p_4} + I_{p_5} = 1 \end{cases}$$

We can find that $\boldsymbol{I}^T$ = (1, 0, 0, 0, 1, 0, 0, 0, 0) and P_a = $\{p_1,\ p_5\}$. Then, we add one token and three tokens to places p_1 and p_5, respectively. We can observe from Table I that after the first iteration step, $\chi_{\gamma_1}(\boldsymbol{M}_0) > b$ holds. Then, we only need to add tokens to γ_1 to decrease the average cycle time. The optimal marking is $\boldsymbol{M}$ = (6, 0, 4, 0, 3, 0, 6, 2, 0)^T and the weight sum of tokens is $f(\boldsymbol{M})$ = 44.

Let us consider Example 2 in Fig. 6. The marking obtained by Eq. (11) is $\boldsymbol{M} = (3, 3, 0, 1, 1)^T$. We have

$$\boldsymbol{M}^\star(p_1) = \left\lfloor \frac{\boldsymbol{M}(p_1)}{gcd_{p_1}} \right\rfloor \cdot gcd_{p_1} = \left\lfloor \frac{3}{2} \right\rfloor \cdot 2 = 2$$

$$\boldsymbol{M}^\star(p_5) = \left\lfloor \frac{\boldsymbol{M}(p_5)}{gcd_{p_5}} \right\rfloor \cdot gcd_{p_5} = \left\lfloor \frac{1}{2} \right\rfloor \cdot 1 = 0$$

Then the initial marking is $\boldsymbol{M}_0 = \boldsymbol{M}^\star = (2, 3, 0, 1, 0)^T$. The iteration process is shown in Table II and the optimal marking is $(2, 3, 4, 1, 0)^T$.

By enumerating all the possible markings for examples in Figs. 5 and 6, we found that the solutions obtained by our proposed approach are optimal.

TABLE IV

A COMPARISON BETWEEN THE APPROACH OF SAUER AND THE COMBINED APPROACH (HE+SAUER).

				He+Sauer [ave]			He+Sauer/Sauer [ave]			He+Sauer/He [ave]		
Nb. of cycles	Nb. of nets	$\|P\|$ [ave]	$\|T\|$ [ave]	Iteration steps	CPU time [s]	Obj. Fun.	Iteration steps	CPU time [s]	Obj. Fun.	Iteration steps	CPU time [s]	Obj. Fun.
1	10	4	4	4.7	23	29.5	38.9%	40.9%	100.0%	157.9%	155.6%	99.1%
2	10	9	8	4.1	63	34.3	18.6%	35.6%	100.0%	215.0%	140.7%	91.4%
4	10	15	12	6.5	193	80.2	3.9%	11.0%	100.0%	189.1%	138.9%	94.1%
6	10	22	17	8.5	472	100.8	2.9%	7.6%	100.0%	293.7%	137.2%	87.9%
10	10	40	31	11.1	973	167.3	11.1/—	973/—	167.3/—	282.0%	131.0%	87.0%

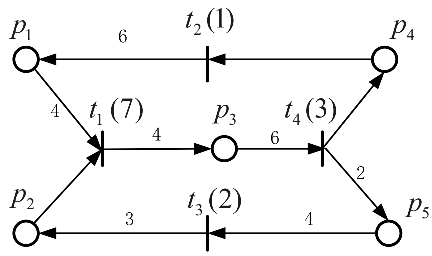


Fig. 6. The TWMG of Example 2 from [7].

D. Comparison with previous approaches

As we know, the previous approach dealing with the marking optimization problem of TWMG is the one presented by Sauer in [7]. We review this iterative heuristic approach in Section III-B and mention that it requires a large number of iterations since it starts from a very large feasible marking.

Adopting the heuristic solution proposed in this section, one starts with a live marking that has a *small* weighted sum. We focus our attention on the low speed circuits whose cycle times are greater than the desired value. To a certain extend, these circuits blind the speed of the system. We never add tokens to circuits whose cycle time is lower than the desired value, i.e., high speed circuits. At every iteration step, we choose one place for each selected circuit by using Eq. (17) and add tokens to it simultaneously. This procedure ensures that the cycle time of the system will decrease to the desired value rapidly. The average cycle time is computed by using simulation which stops when the system enters a cycle (see Section II-D).

In order to compare the approach of Sauer and the proposed approach, we have tested a large number of examples with different net sizes, and for each case we consider a sample of ten nets. All the samples are randomly generated under the assumption that each circuit has at least two places and at most six places. Meanwhile, for each tested example, we initialize $b = max\{x_i \cdot \delta_i, t_i \in T\}$. In the proposed approach, the solution of steps 4 and 8 in Algorithm 1 is computed using Lingo, which takes a negligible time. The highest computational effort is spent in step 6 of Algorithm 1, where we need to determine the cycle time. Similarly, in Sauer's approach, the highest computational effort is due to the repeated computation of the cycle time. Both cases use the Petri net tool HYPENS [24] to compute the cycle time via simulation. The simulation test is executed on a laptop equipped with a 1.8GHZ Core i5 Processor.

The results of a first series of tests are proposed in Table III that shows the comparison between the proposed approach (i.e., He) and that of Sauer. For all cases, we consider the average net size, the average number of iteration steps, the average CPU time, and the average value of obtained objective function. The cardinalities of P and T are approximated to the nearest integer. Note that "—" in Table III means that the computation cannot be finished within a reasonable time. As shown in Table III, we can see that the proposed method is much faster than that by Sauer [7] with the increase of the net size, while the obtained objective function is slightly worse than that of Sauer (i.e., the value of weighted sum $Y^T \cdot \boldsymbol{M}_0$ is greater). The main reason that the proposed approach produces a worse result is that the initial marking computed by Eqs. (11) and (10) does not have the *least* weighted sum to ensure the liveness. Up to now, it is an interesting yet open problem to determine the least live weighted sum of a TWMG. Although we do not allocate any tokens to high speed circuits, the tokens of these circuits may still be too high.

Looking for a better and fast solution, we combine the approach proposed in this paper with that of Sauer [7], namely He+Sauer, as seen in Table IV. First, a candidate marking $\boldsymbol{M}_0$ is computed by the proposed approach. Then we use the approach of Sauer to remove tokens if possible. The simulation results in Table IV present the comparison between the combined approach and the method of Sauer, and also the comparison between the combined approach and the approach proposed in this paper. Comparing the combined approach (He+Sauer) with the approach of Sauer, we always reach the same objective value while the computational costs are significantly reduced.

As one can see, the proposed method needs to find all the elementary circuits and corresponding average cycle times at the first iteration step. Then, we keep track of these slow circuits to allocate tokens. Although in practical examples, the number of circuits in a net is quite reasonable, it is well known that one may define families of nets where the number of circuits can grow exponentially as the net size increases. A case suffering from the circuit explosion is shown in Fig. 7, where Z_i $(i = 1, \cdots, n)$ is an arbitrary integer. The set of circuits of this net is

$$\Gamma = \{p'_1 t_2 p'_2 t_3 \ldots p'_n t_1 \mid (\forall i = 1, \ldots, n)\ p'_i \in \{p_{2i-1}, p_{2i}\}\}$$

and their number is equal to 2^n $(n \geq 2)$. The minimal P-semiflow of each circuit is the characteristic vector of the places along the circuit. Therefore, the sum of all minimal P-semiflows is $Y = Y_1 + Y_2 + \ldots Y_{2^n} = 2^{n-1} \cdot \vec{1}_{2n}$, and we can choose the corresponding P-semiflow $\boldsymbol{y} = \vec{1}_{2n}$ in the criterion $f(\boldsymbol{M}_0)$. Table V shows the simulation results with different number of n and Z_i is a random integer number picked up from the interval $[1, 6]$. As we can see, in the case of $n \geq 6$, the method by Sauer will be more efficient than the proposed method.

Nevertheless, we point out this example is rather academic. In fact, an optimal solution to this problem could be found by studying the equivalent net where places p_2, p_4, $\cdots$, p_{2n} are removed. The equivalent net contains only one circuit, hence can be efficiently studied by the proposed approach. A corresponding optimal solution for the net in Fig. 7 consists in assigning the same number of tokens to the places p_{2i} as in place p_{2i-1}.

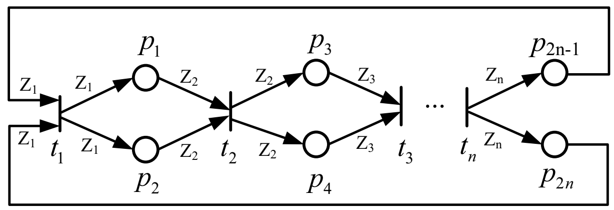


Fig. 7. Example with a large number of circuits.

TABLE V

SIMULATION RESULTS OF THE EXAMPLE IN FIG. 7.

	He/Sauer		
n	Iteration steps	CPU time [s]	Obj. Fun.
2	3/40	29/106	22/22
4	5/66	696/1169	30/30
6	5/104	4916/4399	44/44
7	—/128	—/7320	—/50
8	—/140	—/12194	—/60
9	—/—	—/—	—/—

V. CYCLE TIME OPTIMIZATION FOR TWMGS

A. The main idea of the heuristic solution

In this section, we propose another heuristic solution based on an iterative process to solve the cycle time optimization problem. It starts with a feasible initial marking that makes the net live while the throughput (i.e., the inverse of the cycle time) is low, then we compute the average cycle time of the net and each circuit. It is well known that the cycle time of a TMG can be reduced if and only if tokens are added to the *critical circuit* [5]. Thus, we allocate tokens to the critical circuit at each iteration step as long as the available resources are used up or the cycle time reaches the lower bound. We add tokens to the place that satisfies a certain criterion.

1) A criterion for selecting places: The basic idea of the heuristic is to allocate tokens, which reduces the average cycle time $\chi(\boldsymbol{M})$ as much as possible while increases the cost of resources as less as possible. After we select the

TABLE VI

THE ITERATION PROCESS FOR EXAMPLE 1.

$\boldsymbol{M}_0$	$\chi_{\gamma_1}(\boldsymbol{M}_0)$	$\chi_{\gamma_2}(\boldsymbol{M}_0)$	$\chi(\boldsymbol{M}_0)$	$\chi'(\boldsymbol{M}_0)$	$\gamma^\star$	p	n	$f(\boldsymbol{M})$
$(2, 3, 0, 1, 0)^T$	29	26	30	21	γ_1			20
$(4, 3, 0, 1, 0)^T$	25	26	26	21	γ_2	p_1	2	22
$(4, 4, 0, 1, 0)^T$	25	21	21	21	γ_1	p_2	1	26
$(6, 4, 0, 1, 0)^T$	21	21	21	21		p_1	2	28

initial marking, we can compute the average cycle time $\chi(\boldsymbol{M})$ of the TWMG and the average cycle time $\chi_\gamma(\boldsymbol{M})$ for every elementary circuit. Only when $f(\boldsymbol{M}) < s$ and $\chi(\boldsymbol{M}) > \chi'(\boldsymbol{M})$ hold, i.e., there exist available tokens to be used and the average cycle time does not reach its lower bound, we select the critical circuit $\gamma^\star$ to which tokens should be allocated. If there exists more than one critical circuit, we choose one. At each step, we select one place p and add n tokens to it. The number n is a multiple of gcd_p which represents the minimal number of tokens that we should add to decrease the average cycle time of the critical circuit. It can be computed by using simulation. We denote the decrease in the average cycle time by $\Delta\chi_{\gamma^\star}$ after allocating n tokens to place p. We have

$$\Delta\chi_{\gamma^\star} = \chi_{\gamma^\star}(\boldsymbol{M'}) - \chi_{\gamma^\star}(\boldsymbol{M}) \tag{18}$$

where $\boldsymbol{M'}$ is the marking such that $\boldsymbol{M'}(p) = \boldsymbol{M}(p) + n$ and $\boldsymbol{M'}(p') = \boldsymbol{M}(p')$ if $p' \neq p$. Let g_p be the gain in criterion value, i.e., the resources that we add, where

$$g_p = y_p \cdot n \tag{19}$$

We introduce a criterion Δ_p in which p takes into account both the decreasing of the average cycle time and the gain in criterion value, i.e.,

$$\Delta_p = \frac{g_p}{\Delta\chi_{\gamma^*}} \tag{20}$$

Tokens will be allocated to the place such that

$$\Delta_{p^*} = min\ \Delta_p \tag{21}$$

Note that, the computation of Δ_p is simple: the amount of computation is proportional to the number of places which belong to the critical circuit. At each iteration step, if there is more than one place with minimal value of Δ_p, we keep the optimal allocations that have the minimal average cycle time to next iteration step. Before adding n tokens to the selected place p, we will check if $f(\boldsymbol{M'}) > s$ is true. In case that $f(\boldsymbol{M'}) > s$ holds, another place $p' \in \gamma^\star \cap P\backslash\{p\}$ should be selected instead of place p. We stop the allocation of tokens to the system until all available resources are used up or the average time reaches the lower bound.

TABLE VII

CRITERION VALUES FOR ALL ITERATION STEPS FOR EXAMPLE 1.

$\boldsymbol{M} = (2,3,0,1,0)^T$	p_1	p_3	p_4	$\boldsymbol{M} = (4,3,0,1,0)^T$	p_2	p_3	p_5	$\boldsymbol{M} = (4,4,0,1,0)^T$	p_1	p_3	p_5
n	2	2	1		1	4	2		2	2	1
$\Delta\chi_{\gamma^*}$	4	4	7		5	5	5		4	4	4
g_p	2	4	6		4	8	6		2	4	3
Δ_p	0.5	1	0.85		0.8	1.6	1.2		0.5	1	0.75

2) Heuristic solution: We can summarize the proposed procedure in Algorithm 2.

Algorithm 2: Cycle time optimization

Input: A cyclic TWMG N with a set of elementary circuits Γ, an upper bound s of its weighted sum of tokens, and a P-semiflow $\boldsymbol{y} = \sum_{\gamma \in \Gamma} c_\gamma \cdot \boldsymbol{y}_\gamma$.

Output: An initial marking $\boldsymbol{M}_0$ such that the weighted sum of tokens satisfies $\boldsymbol{y}^T \cdot \boldsymbol{M}_0 \leq s$.

1: Compute the marking $\boldsymbol{M}_D$ as in Eq. (5).

2: For every elementary circuit $\gamma \in \Gamma$, compute $W(\boldsymbol{M}_D^\gamma) = \boldsymbol{y}_\gamma^T \cdot \boldsymbol{M}_D$.

3: For every elementary circuit $\gamma \in \Gamma$, compute $W_L^\gamma = \boldsymbol{y}_\gamma^T \cdot \boldsymbol{M}_D - g$ if possible.

4: Compute a marking $\boldsymbol{M}$ that satisfies Eq. (11).

5: Compute an initial marking $\boldsymbol{M}_0 = \boldsymbol{M}^\star$.

6: Compute the average cycle time $\chi(\boldsymbol{M}_0)$ and $\chi_\gamma(\boldsymbol{M}_0)$, $\forall \gamma \in \Gamma$.

7: If $\chi(\boldsymbol{M}_0) \leq \chi'(\boldsymbol{M})$, stop and $\boldsymbol{M_0}$ is a solution.

8: while $(f(\boldsymbol{M_0}) < s$ and $\chi(\boldsymbol{M_0}) > \chi'(\boldsymbol{M}))$

$\{$

Select a critical circuit $\gamma^\star$;

Compute Δ_p, $\forall p \in \gamma^\star$;

Add tokens to the selected place p, $\boldsymbol{M}_0(p) := \boldsymbol{M}_0(p) + n$;

$\}$

9: Output an initial marking $\boldsymbol{M}_0$.

B. Numerical Examples

In this section, we apply Algorithm 2 to the cycle time optimization of a TWMG and a flexible manufacturing system (FMS).

1) First example: Let us consider the net in Fig. 6. Let $s = 30$ be the maximum number of available resources to be allocated. In Table VI, all optimal allocations are given.

We can find that γ_1 is a critical circuit at the initial marking $\boldsymbol{M}_0$. Accordingly, we compute g_p and $\Delta\chi_{\gamma^*}$ to select the place to which tokens should be added. Table VII shows the criterion values for all iteration steps. After three steps, we stop the iteration since the average cycle time converges to the lower bound. The optimal solution is $\boldsymbol{M}_0 = (6, 4, 0, 1, 0)^T$.

2) Optimization of a flexible manufacturing system: In this subsection, we deal with the example in Fig. 8 from [22]. An FMS composed of three machines $\mathcal{U}_1, \mathcal{U}_2$ and $\mathcal{U}_3$ is considered. It can manufacture two products, denoted by $\mathcal{R}_1$ and $\mathcal{R}_2$. The production mix is 60% and 40% for $\mathcal{R}_1$ and $\mathcal{R}_2$, respectively. The production processes of these products are $\mathcal{R}_1 : \ (\mathcal{U}_1,\ \mathcal{U}_2,\ \mathcal{U}_3)$ and $\mathcal{R}_2 : \ (\mathcal{U}_2,\ \mathcal{U}_1)$.

In this model, there are three types of elementary circuits:

• *Process circuits:* Model the manufacturing process. The tokens belonging to these circuits represent transportation resources.

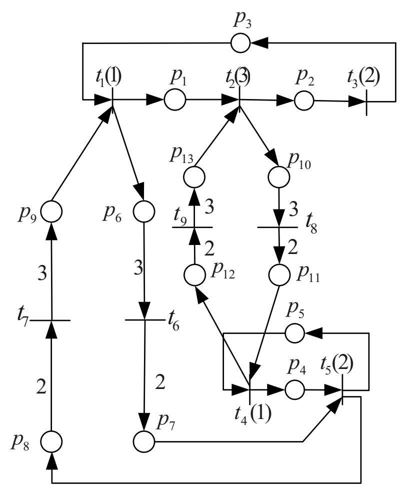


Fig. 8. The TWMG model of a flexible manufacturing system.

- *Command circuits:* Model the control of the system. One command circuit is associated with each machine.
- *Mixed circuits:* Theses circuits are partially composed of parts of the command circuits and parts of the process circuits.

There are six circuits in the FMS:

$$\begin{cases} \gamma_1 = p_1 t_2 p_2 t_3 p_3 t_1 \\ \gamma_2 = p_4 t_5 p_5 t_4 \\ \gamma_3 = p_{10} t_8 p_{11} t_4 p_{12} t_9 p_{13} t_2 \\ \gamma_4 = p_6 t_6 p_7 t_5 p_8 t_7 p_9 t_1 \\ \gamma_5 = p_2 t_3 p_3 t_1 p_6 t_6 p_7 t_5 p_5 t_4 p_{12} t_9 p_{13} t_2 \\ \gamma_6 = p_{10} t_8 p_{11} t_4 p_4 t_5 p_8 t_7 p_9 t_1 p_1 t_2 \end{cases}$$

where γ_1 and γ_2 are process circuits, γ_3 and γ_4 are command circuits, and γ_5 and γ_6 are mixed circuits. The command circuits that model the control of the system must prevent two transitions corresponding to the same machine from being fired simultaneously. Then, they need to satisfy the condition:

$$\boldsymbol{M}(p_{10}) + \boldsymbol{M}(p_{13}) = 3 \ \ \textit{and} \ \ \boldsymbol{M}(p_{11}) + \boldsymbol{M}(p_{12}) = 0 \tag{22}$$

or

$$\boldsymbol{M}(p_{11}) + \boldsymbol{M}(p_{12}) = 2 \ \ \textit{and} \ \ \boldsymbol{M}(p_{10}) + \boldsymbol{M}(p_{13}) = 0 \tag{23}$$

For the command circuit γ_3 in Fig. 8, the conservative component is:

$$2\boldsymbol{M}(p_{10}) + 3\boldsymbol{M}(p_{11}) + 3\boldsymbol{M}(p_{12}) + 2\boldsymbol{M}(p_{13}) = 6 \tag{24}$$

and this command circuit cannot be allocated tokens any more.

TABLE VIII

THE ITERATION PROCESS FOR THE FMS.

M_0	χ_{γ_1}	χ_{γ_2}	χ_{γ_3}	χ_{γ_4}	χ_{γ_5}	χ_{γ_6}	χ	χ'	$\gamma^\star$	p	n	$f(M_0)$
(1,0,0,1,0,0,1,1,1,0,0,2,0)^T	18	6	11	7	15	9	21	11	γ_1			35
(1,1,0,1,0,0,1,1,1,0,0,2,0)^T	9	6	11	7	12	9	13	11	γ_5	p_2	1	38
(1,0,1,1,0,0,1,1,1,0,0,2,0)^T	9	6	11	7	12	9	13	11	γ_5	p_3	1	38
(1,1,0,1,1,0,1,1,1,0,0,2,0)^T	9	4	11	7	9	9	11	11		p_5	1	42
(1,0,1,1,1,0,1,1,1,0,0,2,0)^T	9	4	11	7	9	9	11	11		p_5	1	42

The minimal T-semiflows of γ_1, γ_2, γ_3, γ_4, γ_5, and γ_6 are

$$\begin{cases} \boldsymbol{x}_1 = (1,1,1,0,0,0,0,0,0)^T \\ \boldsymbol{x}_2 = (0,0,0,1,1,0,0,0,0)^T \\ \boldsymbol{x}_3 = (0,3,0,2,0,0,0,1,1)^T \\ \boldsymbol{x}_4 = (3,0,0,0,2,1,1,0,0)^T \\ \boldsymbol{x}_5 = (3,3,3,2,2,1,0,0,1)^T \\ \boldsymbol{x}_6 = (3,3,0,2,2,0,1,1,0)^T \end{cases}$$

and the minimal P-semiflows of γ_1, γ_2, γ_3, γ_4, γ_5 and γ_6 are

$$\begin{cases} \boldsymbol{y}_1 = (1,1,1,0,0,0,0,0,0,0,0,0,0)^T \\ \boldsymbol{y}_2 = (0,0,0,1,1,0,0,0,0,0,0,0,0)^T \\ \boldsymbol{y}_3 = (0,0,0,0,0,0,0,0,0,2,3,3,2)^T \\ \boldsymbol{y}_4 = (0,0,0,0,0,2,3,3,2,0,0,0,0)^T \\ \boldsymbol{y}_5 = (0,2,2,0,3,2,3,0,0,0,0,3,2)^T \\ \boldsymbol{y}_6 = (2,0,0,3,0,0,0,3,2,2,3,0,0)^T \end{cases}$$

We assume that $c_\gamma = 1$, $\forall \gamma \in \Gamma$. Therefore, the P-semiflow used in the performance index is $\boldsymbol{y} = \boldsymbol{y}_1 + \boldsymbol{y}_2 + \boldsymbol{y}_3 + \boldsymbol{y}_4 + \boldsymbol{y}_5 + \boldsymbol{y}_6$= (3, 3, 3, 4, 4, 4, 6, 6, 4, 4, 6, 6, 4)^T, and the minimal T-semiflow of the net is $\boldsymbol{x}$=(3, 3, 3, 2, 2, 1, 1, 1, 1)^T.

Note that, both process circuits γ_1 and γ_2 are timed marked graphs. We only need put one token in each circuit initially. We have

$$\begin{cases} \gamma_4 : W_L = 6,\ W(\boldsymbol{M}_D) = 7 \\ \gamma_5 : W_L = 6,\ W(\boldsymbol{M}_D) = 7 \\ \gamma_6 : W_L = 6,\ W(\boldsymbol{M}_D) = 7 \end{cases}$$

TABLE IX

SOLUTIONS OBTAINED BY ALGORITHM 2.

Nb. of cycles	Nb. of nets	$\|P\|$ [ave]	$\|T\|$ [ave]	Ave. iteration steps	Ave. CPU time [s]	Ave. obtained cycle time
1	10	4	4	3.6	49	21.4
2	10	9	8	3.4	114	38.8
4	10	15	12	4.3	233	85.2
6	10	22	17	6.1	615	136.1
10	10	40	31	6.6	1121	324.3

and gcd_{p_1} = 1, gcd_{p_2} = 1, gcd_{p_3} = 1, gcd_{p_4} = 1, gcd_{p_5} = 1, gcd_{p_6} = 1, gcd_{p_7} = 1, gcd_{p_8} = 1, gcd_{p_9} = 1, $gcd_{p_{10}}$ = 1, $gcd_{p_{11}}$ = 1, $gcd_{p_{12}}$ = 1, and $gcd_{p_{13}}$ = 1.

min $f(\boldsymbol{M}) = 3\boldsymbol{M}(p_1)+3\boldsymbol{M}(p_2)+3\boldsymbol{M}(p_3)+4\boldsymbol{M}(p_4)+4\boldsymbol{M}(p_5)+4\boldsymbol{M}(p_6)+6\boldsymbol{M}(p_7)+6\boldsymbol{M}(p_8)+4\boldsymbol{M}(p_9)+4\boldsymbol{M}(p_{10}) + 6\boldsymbol{M}(p_{11}) + 6\boldsymbol{M}(p_{12}) + 4\boldsymbol{M}(p_{13})$

$$s.t \begin{cases} \boldsymbol{M}(p_1) + \boldsymbol{M}(p_2) + \boldsymbol{M}(p_3) >= 1 \\ \boldsymbol{M}(p_4) + \boldsymbol{M}(p_5) >= 1 \\ 2\boldsymbol{M}(p_{10}) + 3\boldsymbol{M}(p_{11}) + 3\boldsymbol{M}(p_{12}) + 2\boldsymbol{M}(p_{13}) = 6 \\ 2\boldsymbol{M}(p_6) + 3\boldsymbol{M}(p_7) + 3\boldsymbol{M}(p_8) + 2\boldsymbol{M}(p_9) > 7 \\ 2\boldsymbol{M}(p_2) + 2\boldsymbol{M}(p_3) + 3\boldsymbol{M}(p_5) + 2\boldsymbol{M}(p_6) + 3\boldsymbol{M}(p_7) \\ +3\boldsymbol{M}(p_{12}) + 2\boldsymbol{M}(p_{13}) > 7 \\ 2\boldsymbol{M}(p_1) + 3\boldsymbol{M}(p_4) + 3\boldsymbol{M}(p_8) + 2\boldsymbol{M}(p_9) + 2\boldsymbol{M}(p_{10}) \\ +3\boldsymbol{M}(p_{11}) > 7 \end{cases}$$

Note that in this example, if we use W_L for γ_4, γ_5 and γ_6 to compute an initial marking, there is no feasible solution since the constrains are overly strict. We find the initial marking

$$\boldsymbol{M}_0 = \boldsymbol{M}^\star = (1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 2, 0)^T$$

Let s=45 be the maximum number of available resources that can be allocated. Table VIII shows the iteration process. Both $(1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 2, 0)^T$ and $(1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 2, 0)^T$ are optimal solutions.

We have tested some examples with different net sizes for Algorithm 2 and the simulation results are summarized in Table IX. Note that according to Proposition 4, the number of available resources s should be greater than s^* for each tested net.

VI. Conclusion

This paper deals with determinisitc timed weighted marked graphs. We address the problem of marking optimization and cycle time optimization of a TWMG. The marking optimization problem consists in finding an initial marking to minimize the weighted sum of tokens in places while the average cycle time is less than or equal to a given value. The cycle time optimization problem consists in finding an initial marking to minimize the average cycle time while the weighted sum of tokens in places is less than or equal to a given value. Two heuristic algorithms are proposed to solve these problems.

Numerical simulation studies show that the proposed method in Section IV-B requires less iteration steps and thus is much more efficient than the approach in [7]. In some special cases the objective function obtained may be worse than the one found by Sauer. However, we show that by combining the two approaches, we always reach the same objective function by Sauer [7] with a significant reduction of computational costs. Future work includes finding an analytical solution to compute the average cycle time and proposing optimal solutions to the two optimization problems.

References

[1] E. Teruel, P. Chrzastowski-Wachtel, J.M. Colom, and M. Silva. "On weighted T-Systems," *Appl. Theory Petri Nets*, vol. 616, pp. 348–367, 1992.

[2] M. Nakamura and M. Silva. "Cycle time computation in deterministically timed weighted marked graphs," *in Proc. 7th IEEE Int. Conf. Emerg. Technol. Factory Autom.*, vol. 2, pp. 1037–1046, 1999.

[3] J. Campos, G. Chiola, and M. Silva. "Ergodicity and throughput bounds of Petri nets with unique consistent firing count vector," *IEEE Trans. Software Eng.*, vol. 17, no. 2, pp. 117–125, 1991.

[4] A. Benabid-Najjar, C. Hanen, O. Marchetti, and A. Munier. "Periodic schedules for bounded timed weighted event graphs," *IEEE Trans. Automat. Control*, vol. 57, no. 5, pp. 1222–1232, 2012.

[5] A. Giua, A. Piccaluga, and C. Seatzu. "Firing rate optimization of cyclic timed event graphs," *Automatica*, vol. 38, no. 1, pp. 91–103, 2002.

[6] S. Laftit, J M. Proth, and X. Xie. "Optimisation of invariant criteria for event graphs," *IEEE Trans. Automat. Control*, vol. 37, no. 5, pp. 547–555, 1992.

[7] N. Sauer. "Marking optimization of weighted marked graphs," *Discrete Event Dyn. Syst.*, vol. 13, no. 3, pp. 245–262, 2003.

[8] L. Toursi and N. Sauer. "Branch and bound approach for marking optimization problem of weighted marked graphs," *in Proc. IEEE Int. Conf. Syst., Man, Cybern.*, vol. 2, pp. 1777–1782, 2004.

[9] Z. He, Z.W. Li, and A. Giua. "Marking optimization of deterministic timed weighted marked graphs," *in Proc. 10th IEEE Int. Conf. Autom. Sci. Eng.*, Taipei, Taiwan, August 2014.

[10] J. Campos, G. Chiola, J.M. Colom, and M. Silva. "Properties and performance bounds for timed marked graphs," *IEEE Trans. Fundam. Theory Appl.*, vol. 39, no. 5, pp. 386–401, 1992.

[11] T. Murata. "Petri nets: Properties, analysis and applications," *in Proc. IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

[12] C.V. Ramamoorthy and G.S. Ho. "Performance evaluation of asynchronous concurrent systems using Petri nets," *IEEE Trans. Software Eng.*, vol. 6, no. 5, pp. 440–449, 1980.

[13] C. Ramchandani. "Analysis of asynchronous concurrent systems by timed Petri nets," Ph.D. dissertation, MIT, Cambridge, MA, 1973.

[14] A. Munier. "Régime asymptotique optimal d'un graphe d'événements temporisé généralisé: Application à un problème d'assemblage," *APII*, vol. 27, pp. 487–513, 1993.

[15] P. Chrzastowski-Wachtel and M. Raczunas. "Liveness of weighted circuits and the diophantine problem of frobenius," *Fund. Comput. Theory*, vol. 710, pp. 171–180, 1993.

[16] A.J. Carlier and P. Chrtienne. “Problme dordonnancement: Modlisation, complexit, algorithmes,” Masson, Paris, 1988.

[17] J.V. Millo and R. De Simone. “Periodic scheduling of marked graphs using balanced binary words,” *Theor. Comput. Sci.*, vol. 458, pp. 113–130, 2012.

[18] F. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat. “Synchronization and linearity: An algebra for discrete event systems,” *John Wiley & Sons*, 1992.

[19] D.Y. Chao, M.C. Zhou, and D.T. Wang. “Multiple weighted marked graphs,” *in Proc. 12th IFAC Triennial World Congress*, vol. 4, pp. 259–263, 1993.

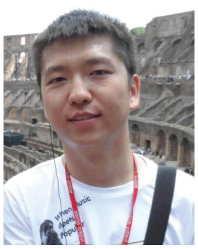
[20] O. Marchetti and A. Munier. “A sufficient condition for the liveness of weighted event graphs,” *Eur. J. Oper. Res.*, vol. 197, no. 2, pp. 532–540, 2009.

[21] O. Marchetti and A. Munier. “Complexity results for weighted timed event graphs,” *Discrete Optim.*, vol. 7, no. 3, pp. 166–180, 2010.

[22] J.M. Proth, N. Sauer, and X. Xie. “Optimization of the number of transportation devices in flexible manufacturing systems using event graphs,” *IEEE Trans. Ind. Electron.*, vol. 44, no. 3, pp. 298–306, 1997.

[23] A.H. Ghamarian, M.C.W. Geilen, S. Stuijk, T. Basten, A.J.M. Moonen, M.J.G. Bekooij, B.D. Theelen, and M.R. Mousavi. “Throughput analysis of synchronous data flow graphs,” *in Proc. 6th IEEE Int. Conf. Appl. Concurrency Syst. Design*, pp. 25–36, 2006.

[24] HYPENS: http://www.diee.unica.it/automatica/hypens/



Zhou He received the B.S. degree in mechanical engineering from Shaanxi University of Science and Technology, Xi’an, China, in 2012. He joined Xidian University, Xi’an, China, in 2012, where is currently pursuing the Ph.D degree in Mechatronic Engineering. He is also a Ph.D student under a joint international supervision agreement with the Information and Systems Science Laboratory (LSIS) of Aix-Marseille University, France. From May 2014 to August 2014, he was a visiting Ph.D student at the Department of Electrical and Electronic Engineering (DIEE) of the University of Cagliari, Italy.

His current research interests focus on control and optimization of timed discrete event systems.

Zhiwu Li (M’06-SM’07) received the B.S., M.S., and Ph.D. degrees in mechanical engineering, automatic control, and manufacturing engineering, respectively, all from Xidian University, Xian, China, in 1989, 1992, and 1995, respectively. He joined Xidian University in 1992 and now he is also with the Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau. Over the past decade, he was a Visiting Professor at the University of Toronto, Technion–Israel Institute of Technology, Martin-Luther University of Halle–Wittenburg, Conservatoire National des Arts et Métiers (CNAM), Meliksah Universitesi. His current research interests include Petri net theory and application, supervisory control of discrete event systems, workflow modeling and analysis, system reconfiguration, game theory, and data and process mining.

He is a member of Discrete Event Systems Technical Committee of the IEEE Systems, Man, and Cybernetics Society, and a member of IFAC Technical Committee on Discrete Event and Hybrid Systems from 2011 to 2014. He serves as a frequent reviewer for 40+ international journals including Automatica and a number of the IEEE Transactions as well as many international conferences. He is listed in Marquis Who’s Who in the world, 27th Edition, 2010. Dr. Li is a recipient of an Alexander von Humboldt Research Grant, Alexander von Humboldt Foundation, Germany. He is a senior member of IEEE and is the founding chair of Xian Chapter of IEEE Systems, Man, and Cybernetics Society.

Alessandro Giua is professor of Automatic Control at the Information and Systems Sciences Laboratory (LSIS) of Aix-Marseille University, France and at the Department of Electrical and Electronic Engineering (DIEE) of the University of Cagliari, Italy. He received a Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, New York in 1992. He has also held visiting positions in several institutions worldwide, including Xidian University, Xian, China.

His research interests include discrete event systems, hybrid systems, networked control systems, Petri nets and failure diagnosis. On these topics he has published extensively and has managed several research projects.

He is the Editor in Chief of the IFAC journal Nonlinear Analysis: Hybrid Systems, and a Department Editor of Discrete Event Dynamic Systems. He has previously served in the editorial board of several journals including the IEEE Trans. on Automatic Control, the IEEE Trans. on Control Systems Technology and the European Journal of Control.

He has been chair of the IFAC Technical Committee 1.3 on Discrete Event and Hybrid Systems from 2008 to 2014 and Chapter Activities chair of the Member Activities Board of the IEEE Control Systems Society from 2006 to 2013. Currently he is serving as a member of IFAC Publications Committee and as a member of the Board of Governors of the IEEE Control Systems Society.

He has been General Chair or Program Chair for international conferences such as the 4th Work. on Discrete Event Systems (1998), the 2nd IFAC Conf. on the Analysis and Design of Hybrid Systems (2006) and the 7th IEEE Int. Conf. on Automation Science and Engineering (2011) and has served in the steering committees of these series. He will be General Chair of the 55th IEEE Conf. on Decision and Control (2016) and Program Chair of the 13th Work. on Discrete Event Systems (2016).