An improved approach for marking optimization of timed weighted marked graphs

Zhou He, Miao Liu, Ziyue Ma, Zhiwu Li, and Alessandro Giua,

Abstract

Timed weighted marked graphs are a mathematical formalism suitable to model automated manufacturing systems in which synchronization and bulk services and arrivals appear, such as assembly lines and kanban systems. In this paper, we aim to develop practically efficient methods for the marking optimization of timed weighted marked graphs, a problem which consists in finding an initial resource assignment to minimize the cost of resources under a given requirement on the cycle time. Starting with a live initial marking, we first compute the critical places of a timed weighted marked graph by exploring an equivalent model called timed marked graph. Then, we develop an analytical method to identify the critical circuit of the system to which tokens will be iteratively added. Application to a real manufacturing system is finally provided, which shows that the developed approach is significantly more efficient than the existing ones.

Published as:

Zhou He, Miao Liu, Ziyue Ma, Zhiwu Li, and A. Giua. "An improved approach for marking optimization of timed weighted marked graphs," *Discrete Event Dynamic Systems*, vol. 29, no. 2, pp.127-143, 2019. DOI: 10.1007/s10626-019-00278-w

Zhou He and Miao Liu are with the College of Mechanical and Electrical Engineering, Shaanxi University of Science and Technology, Xi'an 710021, China, e-mail: hezhou@sust.edu.cn; liumiao19880508@126.com.

Ziyue Ma is with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China, email: maziyue@gmail.com.

Zhiwu Li 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 DIEE, University of Cagliari, Cagliari, Italy, email: giua@diee.unica.it.

I. Introduction

Petri nets (PNs) are a mathematical formalism for modeling, analyzing, and controlling discrete event systems [1]–[4]. Timed marked graphs (TMGs), a class of Timed Petri nets (TPNs) that have been studied since the early 90's, are convenient to model and analyze manufacturing systems such as assembly lines and transfer lines [5].

A generalized model called *timed weighted marked graph* (TWMG) was studied in the literature [6]. This model, characterized by weighted arcs, can conveniently describe systems with batches and assembly/disassembly operations. Analytical methods were proposed to evaluate the cycle time of a TWMG by converting it into an equivalent TMG under different server semantics [7], [10].

To find a trade-off between minimizing the cost of the resources and maximizing the system's throughput, two types of optimization problems were studied in the literature: *cycle time optimization* and *marking optimization*. Giua *et al*. [12] addressed the cycle time optimization of TMGs and proposed different approaches to find an optimal initial marking that minimizes the cycle time of the system, while the cost of resources is smaller than or equal to a given bound. The marking optimization of TMGs was investigated in [11] and a heuristic algorithm was developed to find an initial marking that minimizes the cost of resources, while the cycle time of the system is smaller than or equal to a given value. Nakamura and Silva dealt with the same problem by proposing a sub-optimal two phase approach [9]. First, a reasonable initial marking is computed based on a greedy algorithm. Second, the obtained marking is refined via a tabu-search technique.

Due to the presence of weighted arcs, the optimization problem of TWMGs is much more complicated and has not received much attention in the literature [8]. A heuristic strategy was firstly explored to provide sub-optimal solutions for the cycle time optimization of TWMGs in [13]. We have shown in [15] an analytical technique to compute an optimal marking for the cycle time optimization of TWMGs. First, the (finite) family of TMGs that are equivalent to a given TWMG is characterized, each one valid for a class belonging to a finite partition of the initial markings set. Then, by solving mixed integer linear programming problems (MILPPs), the best solution within each class of TMGs is computed. Finally, the optimal one is selected and the equivalent solution for the original TWMG is computed. However, the computational cost of these optimal approaches is usually very high since the number of classes in the family of equivalent TMGs can grow exponentially with respect to the number of places. For this reason, we believe that sub-optimal approaches based on heuristics, which have also been used in this setting, can still play an important role in many practical applications where due to computational or time constraints an optimal solution cannot be obtained.

Considering the marking optimization of TWMGs, there exist only heuristic approaches in the literature. Sauer [14] developed an iterative heuristic method starting with a feasible marking which contains enough tokens to meet the requirement on cycle time. Then, one place is selected to remove one token until the cycle time increases to its upper bound. It usually needs a large number of iteration steps to remove the additional tokens. He *et al*. [13] proposed a novel heuristic method which was more efficient than that in [14]. It started with a live initial marking whose weighted sum of tokens is small. Then, the critical circuits are searched by simulation and tokens are iteratively added to all of them until the cycle time of the system converges to the upper bound on the cycle time.

The computational bottleneck of the approach proposed in [13] lies in the fact that to find the critical circuit it is necessary to compute the cycle time of each elementary circuit by simulation. The simulation analysis has to explore the dynamic evolutions of the system and inevitably runs into the state explosion problem inherent to discrete models. In the meanwhile, the number of elementary circuits increases exponentially with respect to the net size. Thus, the computational cost of the heuristic methods proposed in [13] and [14] is still not negligible, although by order of magnitude, it is smaller than those of the optimal approaches. As far as we know, it is impossible to extend the optimal techniques used for cycle time optimization in [15] to marking optimization. However, we can use some analytical steps of the optimal approach to further improve the computational efficiency of the heuristic approaches.

It is worthy to mention that synchronous dataflow graphs (SDFGs) are a well-known formalism commonly used to model and analyze data flow applications and real-time embedded streaming applications and are equivalent to TWMGs [16], [17]. The throughput of SDFGs can be computed by transforming an SDFG into an equivalent homogeneous SDFG [18] or by exploring the state space [19]. The buffer minimization problem of SDFGs under a given throughout constraints was investigated in [20], [21].

In this paper, we aim to develop practically efficient methods for marking optimization of TWMGs. An analytical method is originally developed to find the critical circuit of the system, which makes intensive use of linear programming (LP) techniques and avoids enumerating all the elementary circuits and their corresponding cycle time. As a consequence, the most burdensome part of the heuristic approach in [13] can be removed, significantly reducing the computational cost. Extensive results carried out show that this method improves the efficiency of our previous method in [13]. One practical advantage of the developed method is that it can be used in large scale systems.

This paper is organized in six sections. Basic notations of PNs and TWMGS are presented in Section II. Section III gives the problem statement. In Section IV, we present a heuristic approach for the marking optimization problem based on linear algebra. Experimental results are given in Section V. Finally, Section VI draws the conclusions and future work.

II. Background

A. Preliminaries

A *place/transition* net (P/T net) is a four-tuple $N = (P, T, \boldsymbol{Pre}, \boldsymbol{Post})$ with a set of n places P; a set of m transitions T; the pre and post incidence matrix $\boldsymbol{Pre}$, $\boldsymbol{Post} \in \mathbb{N}^{n\times m}$. The *incidence matrix* of a P/T net is denoted by $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$.

A vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_m)^T \in \mathbb{N}^{|T|}$ is a *T-semiflow* if it holds that $\boldsymbol{x} \neq \boldsymbol{0}$ and $\boldsymbol{C} \cdot \boldsymbol{x} = \boldsymbol{0}$. A vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^T \in \mathbb{N}^{|P|}$ is a *P-semiflow* if it holds that $\boldsymbol{y} \neq \boldsymbol{0}$ and $\boldsymbol{y}^T \cdot \boldsymbol{C} = \boldsymbol{0}$. A T-semiflow (resp., P-semiflow) is said to be *minimal* if its support is not a superset of any other T-semiflow's (resp., P-semiflow's) support, and its components are mutually prime.

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

A *weighted marked graph* (WMG) is a subclass of PNs such that each place has single input transition and single output transition, and the weight associated with arcs are positive integer numbers. A timed PN is a pair $N^\delta = (N, \delta)$, where $\delta : T \rightarrow \mathbb{N}$ represents the *firing delay* of transitions [22].

A net is *strongly connected* if there exists a directed path from any node in $P \cup T$ to every other node. A *circuit* of a net is a directed path in the underlying (unweighted) graph that starts and ends with the same node. An *elementary circuit* is a directed path that starts and ends with the same node without passing twice on the same node [23].

Let $t_{in(p)}$ (resp., $t_{out(p)}$) be the unique input (resp., output) transition of place p of a WMG and $w(p)$ (resp., $v(p)$) be the weight of its input (resp., output) arc. We denote by gcd_p the greatest common divisor of $w(p)$ and $v(p)$.

Definition 1: A WMG is said to be *neutral* if for each elementary circuit γ it holds that $\prod_{p \in \gamma} \frac{w(p)}{v(p)} = 1$. □

A strongly connected and neutral WMG is *conservative*, i.e., there exists a P-semiflow whose components are all positive [24], [25]. In addition, there exists a unique minimal T-semiflow $\boldsymbol{x}$ that contains all transitions in its support [26].

The *enabling degree* of a transition t enabled at marking $\boldsymbol{M}$ is the biggest integer number k such that

$$\boldsymbol{M} \geq k \cdot \boldsymbol{Pre}(\cdot, t). \quad (1)$$

From the queue theory point of view, this can be interpreted as the number of servers at each station (transition). In the rest of this paper, we assume that the considered TWMG is strongly connected, neutral, and follows the single server semantics. In another word, each transition represents an operation that can be executed by a single operating unit under the single server semantics. Therefore, regardless of the current enabling degree, a transition may fire at most once at a time. More details can be found in [22], [26].

A *deterministic timed PN* is a pair $N^\delta = (N, \delta)$, where $N = (P, T, \boldsymbol{Pre}, \boldsymbol{Post})$ is a standard PN, and $\delta : T \rightarrow \mathbb{N}$, called delay time, assigns a non-negative integer fixed duration to each transition. In this paper, we will consider deterministic TWMGs.

B. Computation of the cycle time of a TWMG

Definition 2: The *cycle time* $\chi(\boldsymbol{M}_0)$ of a TWMG system $\langle N, \boldsymbol{M_0} \rangle$ is the average time to fire one time the minimal T-semiflow. □

Let γ be a circuit of a TWMG and Γ be the set of elementary circuits of a TWMG. We define $\chi_\gamma(\boldsymbol{M}_0)$ as the cycle time of circuit γ. A *critical circuit* γ_c is a circuit (elementary or not) such that its cycle time is the greatest, i.e., $\chi_{\gamma_c}(\boldsymbol{M}_0) \geq \chi_\gamma(\boldsymbol{M}_0),\ \forall \gamma \in \Gamma$, and places belonging to the critical circuit are called *critical places*.

In a TMG, the cycle time of the net is equal to the cycle time of the slowest elementary circuit: so there always exists a critical elementary circuit and it is sufficient to study only them to identify the bottleneck. On the contrary, in the case of TWMG it may happen that no elementary circuit is critical, thus we need to also study the non-elementary circuits to identify the bottleneck. This makes the analysis of TWMG much more difficult. An example of this is given in the net we will consider in Example 3 where the critical circuit of the TWMG is non-elementary and composed of two elementary circuits.

Simulation can be used to evaluate the cycle time of TWMGs [28]. It has been shown that the execution of a live and strongly connected TWMG is ultimately repetitive [29]. Therefore, the simulation stops when the system enters a cycle and the period of this cycle is a multiple of the cycle time. However, the simulation analysis will explore the dynamic evolution of the system and inevitably runs into the state explosion problem inherent to discrete event models. In the following, we will present an example to illustrate this problem.

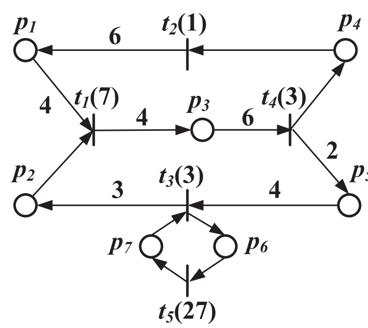

Fig. 1. The TWMG model N considered in Examples 1, 2, 3, and 4.

Example 1. Consider the TWMG model N in Fig. 1. The cycle time of the model with different initial markings $\boldsymbol{M}_0$ was computed by simulation. The obtained values and the corresponding CPU time are given in Table I. The results show that the computational cost of the simulation technique can grow with the cardinality of the state space. ■

TABLE I

CYCLE TIME OF THE NET IN FIG. 1 BY SIMULATION.

Initial marking $\boldsymbol{M}_0$	Cycle time	CPU time [s]
$(2,2,6,0,0,0,1)^T$	31	0.9
$(12,3,12,2,4,1,1)^T$	27	4.9
$(12,13,12,2,4,1,1)^T$	27	80.6
	by transformation	
$(12,13,12,2,4,1,1)^T$	27	2.2

In this paper, we adopt the approach proposed in [7] to transform a TWMG system $\langle N, \boldsymbol{M}\rangle$ into an equivalent

TMG system $\langle \hat{N}, \hat{\boldsymbol{M}} \rangle$ with $\hat{n}$ places and $\hat{m}$ transitions whose cycle time can be rather easily computed by solving an LP as shown in [5]. The computational complexity of the transformation is *pseudo-polynomial* on the net structure (polynomial on $|\boldsymbol{x}|$, where $\boldsymbol{x}$ is the minimal T-semiflow).

Example 2. Consider again the TWMG model N in Fig. 1 whose minimal T-semiflow is $\boldsymbol{x}$=(3, 2, 1, 2, 1)^T. The equivalent intra transition sequential systems of t_1, t_2, t_3, t_4, and t_5, are $\{t_1^1, q_1^1, t_1^2, q_1^2, t_1^3, q_1^3\}$, $\{t_2^1, q_2^1, t_2^2, q_2^2\}$, $\{t_3^1, q_3^1\}$, $\{t_4^1, q_4^1, t_4^2, q_4^2\}$, $\{t_5^1, q_5^1\}$, the equivalent places of $p_1, p_2, p_3, p_4, p_5, p_6$, and p_7 are $\hat{N}(p_1) = \{p_1^1, p_1^2\}$, $\hat{N}(p_2) = \{p_2^1\}$, $\hat{N}(p_3) = \{p_3^1, p_3^2\}$, $\hat{N}(p_4) = \{p_4^1, p_4^2\}$, $\hat{N}(p_5) = \{p_5^1\}$, $\hat{N}(p_6) = \{p_6^1\}$, $\hat{N}(p_7) = \{p_7^1\}$, and the initial marking of the equivalent TMG system is $\hat{\boldsymbol{M}}_0 = (0,0,0,1,0,0,0,0,0,1,0,0,1,0,1,1,0\ ,1,1)^T$.

The cycle time of the TWMG for different initial markings $\boldsymbol{M}_0$ computed by transformation are presented in the second column of Table I. Note that the cycle time is obtained by solving the LP (4) which we will discuss in Section IV-A for the equivalent TMGs. The table shows that the computational cost of this approach does not depend on the number of states, which is the main advantage over the simulation technique. ■

III. Problem statement

The *marking optimization* of a TWMG considered in this paper can be formulated as follows.

$$\begin{aligned} &\min\ \ f(\boldsymbol{M}_0) = \boldsymbol{y}^T \cdot \boldsymbol{M}_0 \\ &\text{s.t. } \chi(\boldsymbol{M}_0) \leq b, \end{aligned} \tag{2}$$

where

- $\chi(\boldsymbol{M}_0)$ is the cycle time of the TWMG system $\langle N, \boldsymbol{M}_0 \rangle$,
- $\boldsymbol{y} = (y_1, \ldots, y_n)^T$ is a non-negative weight vector that represents the cost of the resources, and
- $b \in \mathbb{R}^+$ represents the upper bound on the cycle time which is known, where $\mathbb{R}^+$ denotes the set of positive real numbers[1]

Note that the P-semiflow $\boldsymbol{y}$ in Eq. (2) is chosen as the weighted sum of all minimal P-semiflows, i.e.,

$$\boldsymbol{y} = \sum_{\gamma \in \Gamma} \lambda_\gamma \cdot \boldsymbol{y}_\gamma, \tag{3}$$

where $\boldsymbol{y}_\gamma$ represents the minimal P-semiflow corresponding to circuit γ and λ_γ represents the cost of the resources modeled by tokens in the support of $\boldsymbol{y}_\gamma$ [13].

IV. Marking optimization of TWMGs: an efficient heuristic approach

In this section, we develop a heuristic approach to deal with the marking optimization of TWMGs, which significantly improves the efficiency of the method proposed in [13]. An integer linear programming (ILP) (ILP (11) in [13]) was adopted to compute a live initial marking $\boldsymbol{M}_0$ which has a small weighted sum (for the sake of simplicity, the details are omitted in this paper). Then, we develop an analytical method to compute the critical

[1]As stated in [13], the problem has a solution iff $b \geq \max\{x_i \cdot \delta_i,\ t_i \in T\}$, where x_i is the component of minimal T-semiflow corresponding to transition t_i and δ_i is the fixed delay time of transition t_i.

circuit of the TWMG net which avoids enumerating all the elementary circuits and their corresponding cycle time. Next, one single place in the critical circuit is selected to add tokens. This procedure is repeated until the cycle time of the system is smaller than or equal to its upper bound.

A. Identify the critical circuit of a TWMG

In this paper, we adopt an ILP to evaluate the cycle time of a TWMG and present an analytical approach to determine the critical circuit to which tokens must be added. Starting with a live initial marking $\boldsymbol{M}_0$, we convert the TWMG system $\langle N, \boldsymbol{M}_0\rangle$ into an equivalent TMG system $\langle \hat{N}, \hat{\boldsymbol{M}}_0\rangle$ according to the approach in [7]. The cycle time of the equivalent TMG system can be easily computed as follows.

Theorem 1: (Campos et al. *[5]) The cycle time of a TMG system $\langle \hat{N}, \hat{\boldsymbol{M}}_0\rangle$ can be computed by solving the following LP:*

$$\chi(\hat{\boldsymbol{M}}_0) = \max\ \hat{\boldsymbol{\alpha}}_\gamma^T \cdot \hat{\boldsymbol{Pre}} \cdot \hat{\boldsymbol{\delta}} \quad s.t. \begin{cases} \hat{\boldsymbol{\alpha}}_\gamma^T \cdot \hat{\boldsymbol{C}} = \boldsymbol{0}, & (a) \\ \hat{\boldsymbol{\alpha}}_\gamma^T \cdot \hat{\boldsymbol{M}}_0 = 1, & (b) \\ \hat{\boldsymbol{\alpha}}_\gamma \geq \boldsymbol{0}, & (c) \end{cases} \tag{4}$$

where $\hat{\boldsymbol{\delta}} \in \mathbb{R}^{\hat{m}\times 1}$ is the firing delay vector. The three constraints in (4) *enforce the vector of decision variable $\hat{\boldsymbol{\alpha}}_\gamma \in \mathbb{R}^{\hat{n}\times 1}$ to be a P-semiflow that represents characteristic vector of the places along the slowest circuit.* ■

We denote the set of critical places of the TMG system and the TWMG system by $\hat{P}_c$ and P_c, respectively. In the following proposition and remark, we show how to compute the critical circuit of a TWMG.

Proposition 1: Let $\langle \hat{N}, \hat{\boldsymbol{M}}_0\rangle$ be a TMG system equivalent to a TWMG system $\langle N, \boldsymbol{M}_0\rangle$ according to the approach in [7], and $\hat{\boldsymbol{\alpha}}_\gamma$ be the vector computed by solving LP (4)*. Then, the set of critical places of the TWMG system $\langle N, \boldsymbol{M}_0\rangle$ can be computed by*

$$P_c = \{p_i | \hat{\alpha}_\gamma(p_i^j) > 0,\ i = 1, \ldots, n,\ j = 1, \ldots, n_i\ \}. \tag{5}$$

Proof: According to constraint (a) in Eq. (4), $\hat{\boldsymbol{\alpha}}_\gamma$ is also a P-semiflow of the equivalent TMG (but for a scalar factor). As the TMGs and TWMGs considered in this paper are strongly connected nets, a P-semiflow corresponds to a circuit of the net. Thus, the support of $\hat{\boldsymbol{\alpha}}_\gamma$ represents the set of critical places of the TMG system $\langle \hat{N}, \hat{\boldsymbol{M}}_0\rangle$, i.e.,

$$\hat{P}_c = \{p | \hat{\alpha}_\gamma(p) > 0\}. \tag{6}$$

It has been proved in [7] that both the PN language and the the critical time of the two systems are identical, i.e., the critical circuit $\hat{\gamma}_c$ of the TMG system preserves the timing information of the critical circuit γ_c (elementary or not) of the TWMG system and $\chi_{\hat{\gamma}_c}(\hat{\boldsymbol{M}}_0) = \chi_{\gamma_c}(\boldsymbol{M}_0)$.

According to the approach in [7], places p_i^j $(i = 1, \ldots, n,\ j = 1, \ldots, n_i)$ of the TMG are equivalent to place $p_i \in P$ of the TWMG and places $q's$ that belong to the intra transition sequential systems are generated by transformation of transitions of the TWMG. Thus, the set of critical places of the TWMG system $\langle N, \boldsymbol{M}_0 \rangle$ can be computed by Eq. (5). ■

Remark 1: The critical circuit γ_c that corresponds to the set of critical places P_c is unique due to the fact that each place of the net has only one input arc and only one output arc. ■

Example 3. Let us consider the TWMG model N depicted in Fig. 1 with an initial marking $\boldsymbol{M}_0 = (2, 2, 6, 0, 0, 0, 1)^T$. It consists of three elementary circuits: $\gamma_1 = \{p_1, t_1, p_3, t_4, p_4, t_2\}$, $\gamma_2 = \{p_2, t_1, p_3, t_4, p_5, t_3\}$, and $\gamma_3 = \{p_6, t_5, p_7, t_3\}$.

By solving LP (4) for the equivalent TMG system $\langle \hat{N}, \hat{\boldsymbol{M}}_0 \rangle$, we obtain the cycle time $\chi(\hat{\boldsymbol{M}}_0) = 31$ and $\hat{\boldsymbol{\alpha}}_\gamma = (1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, \ 0, 0, 0, 0, 0, 0)^T$. The critical circuit of the TMG system is

$$\hat{\gamma}_c = \{p_1^1, t_1^1, q_1^1, t_1^2, p_3^2, t_4^2, p_5^1, t_3^1, p_2^1, t_1^3, p_3^1, t_4^1, p_4^1, t_2^1\}.$$

According to Proposition 1, we have $P_c = \{p_1, p_2, p_3, p_4, p_5\}$. Obviously, the critical circuit of the TWMG system is

$$\gamma_c = \gamma_1 \cup \gamma_2 = \{p_1, t_1, p_3, t_4, p_4, t_2, p_2, t_1, p_3, t_4, p_5, t_3\}.$$

From Table II, one can easily find that the cycle time of γ_3 is the maximal one among all elementary circuits, i.e., γ_3 is the slowest elementary circuit. Nevertheless, in this example the critical circuit γ_c is composed of γ_1 and γ_2. ■

TABLE II

CYCLE TIME OF THE NET AND EACH ELEMENTARY CIRCUIT.

$\boldsymbol{M}_0$	$\chi(\boldsymbol{M}_0)$	$\chi_{\gamma_1}(\boldsymbol{M}_0)$	$\chi_{\gamma_2}(\boldsymbol{M}_0)$	$\chi_{\gamma_3}(\boldsymbol{M}_0)$
$(2, 2, 6, 0, 0, 0, 1)^T$	31	29	27	30

B. Select a place to add tokens

In our previous work [13], after selecting an initial marking $\boldsymbol{M}_0$, we added tokens to some elementary circuits as follows:

1) If there exists any elementary circuit γ whose cycle time is greater than b, i.e., $\chi_\gamma(\boldsymbol{M}_0) > b$, then tokens will be added to all of them.
2) If $\chi(\boldsymbol{M}_0) > b$ and there does not exist elementary circuit γ that has cycle time greater than b, then tokens will be added to the elementary circuits whose cycle time is the maximal one.

However, we find that the method proposed in [13] may add redundant (additional) tokens, as shown in the following example.

Example 4. Let us consider again the TWMG model N depicted in Fig. 1 with an initial marking $\boldsymbol{M}_0 = (2, 2, 6, 0, 0, 0, 1)^T$ and we assume that the upper bound on the cycle time is $b = 30$.

According to the method proposed in [13], tokens will be firstly added to the elementary circuit γ_3 instead of the critical circuit γ_c which is composed of γ_1 and γ_2. Obviously, the cycle time of the system does not decrease since it is not constrained by γ_3. In fact, to decrease the cycle time of the system to the desired value b, it is sufficient to add two tokens to place p_2. Thus, tokens added to γ_3 are redundant. ■

In this paper, we add tokens to the critical circuit γ_c that constrains the cycle time of the system. One place which uses the minimal cost of resources of the critical circuit γ_c is selected, i.e.,

$$\min_{p \in \gamma_c} \text{gcd}_p \cdot y_p \tag{7}$$

where gcd_p is the greatest common divisor of the input arc weight and the output arc weight of place p and $\boldsymbol{y} = (y_1, \ldots, y_n)$ is a weighted vector as defined in Eq. (3).

Combining the methods proposed in Sections IV-A and IV-B, we can improve the heuristic method in [13] as follows.

Algorithm 1 An improved heuristic marking optimization approach

1: **Input:** A TWMG N, a cycle time upper bound b, and a P-semiflow $\boldsymbol{y}$.
2: **Output:** An initial marking $\boldsymbol{M}_0$ such that $\chi(\boldsymbol{M}_0) \leq b$.
3: Compute a live initial marking $\boldsymbol{M}_0$.
4: Compute the cycle time $\chi(\boldsymbol{M}_0)$ and the critical circuit γ_c by Proposition 1.
5: **while** $\chi(\boldsymbol{M}_0) > b$ **do**
6:     Find the critical circuit γ_c according to Proposition 1;
7:     Compute a place p according to Eq. (7);
8:     $M_0(p) := M_0(p) + \text{gcd}_p$;
9:     Update $\chi(\boldsymbol{M}_0)$.
10: **end while**
11: Return $\boldsymbol{M}_0$.

Proposition 2: The output of Algorithm 1 can provide a suboptimal solution of the marking optimization problem as defined in Eq. (2).

Proof: Step 3 of Algorithm 1 guarantees that the heuristic approach starts with an initial marking $\boldsymbol{M}_0$ whose cycle time is finite. During the heuristic greedy process (while loop in Steps 4-11), the critical circuit is searched and tokens will be iteratively added.

For an arbitrary iteration step i, let $\gamma_{c,i}$ be the critical circuit in the i-th iteration. After adding gcd_p tokens to place p that belongs to $\gamma_{c,i}$, we obtain a new initial marking $\boldsymbol{M}_{0,i+1}$ and identify its critical circuit $\gamma_{c,i+1}$, there exist three possible situations:

S1): $\chi(\boldsymbol{M}_{0,i+1}) < \chi(\boldsymbol{M}_{0,i})$;

In such a case, the cycle time of net has decreased.

S2): $\chi(\boldsymbol{M}_{0,i+1}) = \chi(\boldsymbol{M}_{0,i})$ and $\gamma_{c,i+1} = \gamma_{c,i}$;

In such a case, the cycle time of the net and the critical circuit remain the same. However, in a finite step k ($k > i$) when we put enough tokens to the critical circuit $\gamma_{c,i}$, the cycle time of $\gamma_{c,i+1}$ will eventually decrease and we have a new critical circuit $\gamma_{c,k}$. If the cycle time of the net does not decrease, then it goes to *S3*.

S3): $\chi(\boldsymbol{M}_{0,k}) = \chi(\boldsymbol{M}_{0,i})$ and $\gamma_{c,k} \neq \gamma_{c,i}$;

In such a case, there exits more than one critical circuit associated to $\boldsymbol{M}_{0,i}$. However, due to the fact that the TWMG considered has a finite number of circuits, all the critical circuits associated to $\boldsymbol{M}_{0,i}$ will ultimately be non-critical by adding enough tokens. Then, it goes to *S1*.

As a conclusion, Algorithm 1 guarantees that an initial marking $\boldsymbol{M}_0$ whose cycle time is smaller than or equal to the desired value b can be ultimately obtained, which provides a feasible solution for the marking optimization problem as defined in Eq. (2). ■

We conclude this section with a brief discussion on the complexity of the proposed approach. The complexity of the algorithm in [7] to obtain the cycle time of a TWMG is polynomial with respect to the minimal T-semiflow, i.e., $O(|\boldsymbol{x}|)$, and the complexity of the developed algorithm to find the critical circuit γ_c is polynomial with respect to the number of places of the equivalent TMG, i.e., $O(|\hat{n}|)$. At each iteration step of Algorithm 1, a place $p \in \gamma_c$ is selected to add at least one token. To ensure the liveness of the TWMG, the initial marking obtained by using ILP (11) in [13] should contain at least one token. Under single server semantics, a marking $\boldsymbol{M}_f$ which contains enough tokens for each places is feasible for problem (2), where

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

Hence, Algorithm 1 terminates after at most $\sum\limits_{p \in P} M_f(p) - 1$ iterations.

V. Experimental results

In this section, the heuristic approach proposed in Section IV is applied to a real manufacturing system that is a hydraulic torque converter assembly line. Then, the comparison with the previous approaches are discussed in Section V-B.

A. Application to a real manufacturing system

In this subsection, we illustrate the developed method by studying a hydraulic torque converter assembly line in a car component manufacturing plant in Xi'an, China. The hydraulic torque converter (denoted by $\mathcal{P}_0$) is assembled by four different types of products (denoted by $\mathcal{P}_1, \ldots, \mathcal{P}_4$) that are pump wheel, turbine, guide wheel, and cover wheel. The system consists of 17 different types of machines. After the hydraulic torque converter is assembled, it is unloaded by an automated guided vehicle (AGV). The production processes of these products are shown in Table III. Note that the turbine ($\mathcal{P}_2$) is also assembled by two different parts that are denoted by $\mathcal{P}_{2,1}$ and $\mathcal{P}_{2,2}$.

The TWMG model of the assembly line is shown in Fig. 2 and consists of 41 places and 25 transitions. The total number of elementary circuits is 109, which is composed of three different types of elementary circuits:

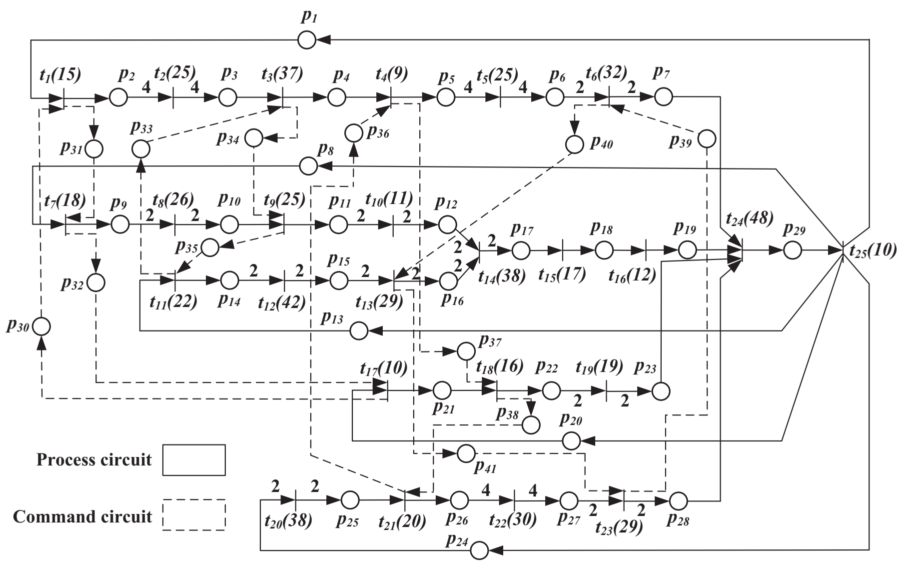

Fig. 2. The TWMG model of the hydraulic torque converter assembly line.

TABLE III

THE PRODUCTION PROCESS OF THE HYDRAULIC TORQUE CONVERTER ASSEMBLY LINE.

Products	Production process
Pump wheel ($\mathcal{P}_1$)	$\mathcal{M}_1(15) \rightarrow \mathcal{M}_2(25) \rightarrow \mathcal{M}_3(37) \rightarrow \mathcal{M}_4(9)$ $\rightarrow \mathcal{M}_5(25) \rightarrow \mathcal{M}_6(32)$
Turbine ($\mathcal{P}_2$)	$\mathcal{P}_{2,1}$: $\mathcal{M}_1(18) \rightarrow \mathcal{M}_7(26) \rightarrow \mathcal{M}_3(25) \rightarrow \mathcal{M}_8(11)$ $\rightarrow \boldsymbol{\mathcal{M}_{10}(38)} \rightarrow \boldsymbol{\mathcal{M}_{11}(17)} \rightarrow \boldsymbol{\mathcal{M}_{12}(12)}$ $\mathcal{P}_{2,2}$: $\mathcal{M}_3(22) \rightarrow \mathcal{M}_9(42) \rightarrow \mathcal{M}_6(29)$ $\rightarrow \boldsymbol{\mathcal{M}_{10}(38)} \rightarrow \boldsymbol{\mathcal{M}_{11}(17)} \rightarrow \boldsymbol{\mathcal{M}_{12}(12)}$
Guide wheel ($\mathcal{P}_3$)	$\mathcal{M}_1(10) \rightarrow \mathcal{M}_4(16) \rightarrow \mathcal{M}_{13}(19)$
Cover wheel ($\mathcal{P}_4$)	$\mathcal{M}_{14}(38) \rightarrow \mathcal{M}_4(20) \rightarrow \mathcal{M}_{15}(30) \rightarrow \mathcal{M}_6(29)$
Hydraulic torque converter ($\mathcal{P}_0$)	$\mathcal{M}_{16}(48) \rightarrow \mathcal{M}_{17}(10)$

1) First is the *process circuit* γ_{pr} which models the manufacturing process. For instance, circuit $\{p_1, t_1, p_2, t_2, p_3, t_3, p_4, t_4, p_5, t_5, p_6, t_6, p_7, t_{24}, p_{29}, t_{25}\}$ is a process circuit which represents the manufacturing process of the pump wheel $\mathcal{P}_1$. This type of circuits represents the physical part of the system, and the tokens belonging to these circuits represent pallets for products.
2) Second is the *command circuit* γ_{com} which represents the control strategy of the system. One command circuit is associated with each machine to ensure that they are cyclically used in both processes and each command circuit is a TMG. For instance, circuit $\{p_{30}, t_1, p_{31}, t_7, p_{32}, t_{17}\}$ is a command circuit. To prevent two transitions corresponding to the same machine fire simultaneously, each command circuit contains only

one token.

3) Third is the *mixed circuit* γ_{mix} which consists of both the command circuits and the process circuits. For instance, circuit $\{p_8, t_7, p_{32}, t_{17}, p_{21}, t_{18}, p_{22},\ t_{19}, p_{23}, t_{24}, p_{29}, t_{25}\}$ is a mixed one.

The raw material of each production line is assumed to be sufficient and the problem is to optimize the total cost of the pallet due to the inventory, i.e., the semi-processed pieces that are contained in system. Tokens belonging to the command circuits represent the logic control of the system and do not have a cost. Therefore, the P-semiflow used in the objective function is selected as $\boldsymbol{y} = \sum \boldsymbol{y}_{\gamma_{pr}}$.

TABLE IV

SIMULATION RESULTS OF THE HYDRAULIC TORQUE CONVERTER ASSEMBLY LINE.

Iteration step	$\boldsymbol{M}_0$	$\chi(\boldsymbol{M}_0)$	b	$\|P_c\|$	Selected place
0	$8p_1 + 4p_8 + 4p_{13} + 4p_{20} + 6p_{24}$ $+p_{30} + p_{33} + p_{36} + p_{39}$	673	336	17	p_2
⋮	⋮	⋮	⋮	⋮	⋮
10	$8p_1 + 5p_8 + 5p_{13} + 5p_{20} + 6p_{24}$ $+p_{30} + p_{33} + p_{36} + p_{39} + 3p_2$ $+2p_4 + p_5 + p_9$	336	336	3	
				$f(\boldsymbol{M}_0)$	36
				CPU time [s]	133

For the application of Algorithm 1, MATLAB has been used with software package Gurobi [30]. A live initial marking is first computed which can be expressed as $\boldsymbol{M}_0 = 8p_1 + 4p_8 + 4p_{13} + 4p_{20} + 6p_{24} + p_{30} + p_{33} + p_{36} + p_{39}$, i.e., eight tokens in p_1, four tokens in each of p_8, p_{13}, and p_{20}, six tokens in p_{24}, and one token in each of p_{30}, p_{33}, p_{36}, and p_{39}. The cycle time of this marking is $\chi(\boldsymbol{M}_0) = 673$ which is greater than the upper bound on the cycle time $b = 336$.

Note that we cannot select places that belong to the command circuits to add tokens. After ten iteration steps, an initial marking is obtain whose cycle time is equal to the upper bound b as shown in Table IV.

B. Comparison with previous approaches

In [13] and [14], the authors resorted to simulation for computing the cycle time of TWMGs. In order to select a place which has redundant (unused) tokens that can be removed without any influence on the cycle time, the method presented in [14] needs to simulate the system until the estimation of the cycle time of each transition converges to a preassigned precision. He *et al.* [13] compute the cycle time of TWMGs by simulating the system until its dynamic behavior enters a cycle, i.e., the current state of the system is identical with a previous one. In order to find all the critical circuits, the proposed approach needs to compute the cycle time for every elementary circuit whose computational cost can grow exponentially with respect to the net size.

To better compare the efficiency of the proposed approach in this paper with the approaches developed in [13] and [14], we test some cases taken from the literature. Cases 1 and 2 are taken from [14] and [27], respectively, Case 3 is a flexible manufacturing system studied in [10], and Case 4 is the hydraulic torque converter assembly line depicted in Fig. 2 in Section V-A. For the application of Algorithm 1, MATLAB has been used with Gurobi, and the simulation based methods in [13] and [14] are implemented by MATLAB with the PN tool HYPENS [28] on a computer with i7 processor 3.60 GHz and 8 GB memory. In Table V, we show the number of places

TABLE V

A COMPARISON BETWEEN THE APPROACH OF SAUER [14], HE *et. al* [13], AND THE APPROACH PROPOSED IN THIS PAPER.

Sauer [14]						
Case	$\|P\|$	$\|T\|$	Nb. of circuits	Iteration steps	CPU time [s]	Objective function
1	5	4	2	17	12	28
2	9	6	4	65	175	32
3	24	12	42	85	610	307
4	41	25	109	o.o.t.	o.o.t.	o.o.t.
He *et. al* [13]						
Case	$\|P\|$	$\|T\|$	Nb. of circuits	Iteration steps	CPU time [s]	Objective function
1	5	4	2	2	7	28
2	9	6	4	2	27	32
3	24	12	42	3 (7)	265 (329)	337 (307)
4	41	25	109	o.o.t.	o.o.t.	o.o.t.
Proposed approach						
Case	$\|P\|$	$\|T\|$	Nb. of circuits	Iteration steps	CPU time [s]	Objective function
1	5	4	2	2	6	28
2	9	6	4	2	10	32
3	24	12	42	3 (5)	19 (51)	311 (307)
4	41	25	109	11 (17)	133 (639)	36 (30)

and transitions, the number of circuits, the number of iteration steps, the CPU time, and the value of obtained objective function for each case. Note that "o.o.t" (out of time) means that the solution cannot be found within a reasonable time. For Cases 1 and 2, all the approaches can provide optimal solutions. For Case 3, the obtained objective function of the approach in [13] and our approach are worse than that of Sauer. Nevertheless, the obtained solution can be further improved by employing Sauer's method, i.e., when stoping adding tokens to the system, we can adopt the approach of Sauer to remove tokens if possible. As one can see the results in parenthesis in Table V, by combining Sauer's approach, the obtained objective functions of the approach in [13] and our approach are decreased.

From the experimental results, we can conclude that the proposed approach greatly improves the efficiency of our previous method [13] by taking the advantage of LP technique to avoid enumerating all the elementary circuits and their corresponding cycle times. However, when comparing our results with previous methods in [13] and [14], it must be pointed out that all of these approaches are based on heuristic strategy. Therefore, the optimal solution cannot be guaranteed. To the best of our knowledge, there does not exist an optimal method for marking optimization of TWMGs. In addition, the lower bound estimation of the cost of resources to reach a desired cycle time is still an open issue.

VI. Conclusion

In this paper, we develop practically efficient method for marking optimization of deterministic TWMGs, which consists in finding an initial resource assignment to minimize the cost of resources under a given requirement on the cycle time. The proposed heuristic approach makes intensive use of LP technique to find the critical circuit and avoids enumerating all the elementary circuits and their corresponding cycle time. As a consequence, the most burdensome part of the approach in [13] can be removed which significantly reduces the computational cost. The method presented in this paper is applicable to the infinite server case by using the results of [25]. Our future work focuses on the generalization to a larger class of models, such as timed weighted marked graphs with choices or nets with monitor places.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant Nos. 61803246, 61472295, and 61703321, the Science and Technology Development Fund, MSAR, under Grant No. 078/2015/A3, and the Natural Science Foundation of Shaanxi University of Science and Technology under Grant Nos. 2016BJ-15 and 2017BJ-39.

References

[1] Z. Y. Ma, Z. W. Li, and A. Giua. "Design of optimal Petri net controllers for disjunctive generalized mutual exclusion constraints," *IEEE Trans. Autom. Control*, vol. 60, no. 7, pp. 1774–1785, 2015.

[2] Y. Tong, Z.W. Li, C. Seatzu, A. Giua. "Current-state opacity enforcement in discrete event systems under incomparable observations," *Discrete Event Dyn. Syst.*, vol. 28, no. 2, pp. 161–182, 2018.

[3] M. Liu, S. G. Wang, M. C. Zhou, D. Liu, A. Al-Ahmari, N. Q. Wu, and Z. W. Li. "Deadlock and liveness characterization for a class of generalized Petri nets," *Infor. Sci.*, vol. 420, pp. 403–416, 2017.

[4] Z. Y. Ma, Z. W. Li, and A. Giua. "Characterization of admissible marking sets in Petri nets with conflicts and synchronizations," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1329–1341, 2017.

[5] 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.

[6] B. Cottenceau, L. Hardouin, and J. L. Boimond. "Modeling and control of weight-balanced timed event graphs in dioids," *IEEE Trans. Autom. Control*, vol. 59, no. 5, pp. 1219–1231, 2014.

[7] A. Munier. "Régime asymptotique optimal d'un graphe d'événements temporisé généralisé: Application à un problème d'assemblage," *RAIRO-APII*, vol. 27, pp. 487–513, 1992.

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

[9] M. Nakamura and M. Silva. "An iterative linear relaxation and tabu search approach to minimum initial marking problems of timed marked graphs," *in Proc. European control conf.,* pp. 985–990, 1999.

[10] 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.

[11] S. Lafit, J. M. Proth, and X. L. Xie. "Marking optimization in timed event graphs," *in Proc. Int Conf. Appl. Theory Petri Nets*, Springer, Berlin Heidelberg, pp. 281–300, 1991.

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

[13] Z. He, Z. W. Li, and A. Giua. "Optimization of deterministic timed weighted marked graphs," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 1084–1095, 2017.

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

[15] Z. He, Z. W. Li, and A. Giua. "Cycle time optimization of deterministic timed weighted marked graphs by transformation," *IEEE Trans. Control Syst. Technol.*, vol.25, no. 4, pp. 1318–1330, 2017.

[16] E. Lee and D. Messerschmitt. "Synchronous data flow," *in Proc. IEEE*, vol. 75, no. 9, pp. 1235–1245, 1987.

[17] S. Bhattacharyya, P. Murthy, and E. Lee. "Synthesis of embedded software from synchronous dataflow specifications," *J. VLSI signal processing*, vol. 21, no. 2, pp. 151–166, 1999.

[18] S. Sriram and S. S. Bhattacharyya. Embedded multiprocessors: Scheduling and synchronization. Boca Raton, CRC press, 2009.

[19] 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 graph," *in Proc. 6th Int. Appl. Concur. Syst. Design*, pp. 25–36, 2006.

[20] R. Govindarajan and G. R. Gao. "Rate-optimal schedule for multi-rate dsp computations," *J. VLSI signal processing*, vol. 9, no. 3, pp. 211–232, 1995.

[21] S. Stuijk, M. Geilen, and T. Basten. "Exploring trade-offs in buffer requirements and throughput constraints for synchronous dataflow graphs," *in Proc. 43rd Design Autom. Conf.*, pp. 899–904, 2006.

[22] J. H. van Schuppen, M. Silva, and C. Seatzu. Control of discrete-event systems-automata and Petri net perspectives, *Lect. Notes Control Inf. Sci.*, Springer, vol. 433, pp. 319–340, 2012.

[23] Y. Tong, Z. W. Li, C. Seatzu, and A. Giua. "On the equivalence of observation structures for Petri net generators," *IEEE Trans. Autom. Control*, vol. 61, no. 9, pp. 2448–2462, 2016.

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

[25] Z. He, Z.W. Li, and A. Giua, "Performance optimization for timed weighted marked graphs under infinite server semantics," *IEEE Trans. Autom. Control*, vol. 63, no. 8, pp. 2573–2580, 2018.

[26] 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.

[27] 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.*, pp. 413–418, Taipei, Taiwan, August 2014.

[28] F. Sessego, A. Giua, and C. Seatzu. "HYPENS: a Matlab tool for timed discrete, continuous and hybrid Petri nets," *in Proc. Int. Conf. Appl. Theor. Petri Nets*, pp. 419–428, 2008.

[29] 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.

[30] Optimization Gurobi. [Online]. Available: http://www.gurobi.com/.