确定性时延**Petri**网带权标记图的性能优化

作者姓名 何舟

指导教师姓名、职称 李志武 教授

第二指导教师 **Alessandro GIUA** 教授

申请学位类别 工学博士

Performance Optimization of a Class of Deterministic Timed Petri Nets: Weighted Marked Graphs

A Doctoral Dissertation
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in Mechatronic Engineering at
XIDIAN UNIVERSITY

and

in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in Automatique at
AIX-MARSEILLE UNIVERSITY

By

Zhou HE

Supervisor: Zhiwu LI Professor

The Second Supervisor: Alessandro GIUA Professor

April 2017

摘要

在过去的几十年里，越来越多的人意识到运用形式化方法来解决工业领域内有关监督与可靠性、故障诊断、及资源最优化利用等问题的重要性。其中，自动制造系统由于能够满足不断变化需求的复杂工艺周期而显得尤为重要。现代自动制造系统通常是由数控机床、组装台、自动导航车辆、机器人、传送带以及计算机控制系统等若干相互关联的子系统构成。制造商利用自动化机器和控制器来快速、高效地生产出优质产品。与此同时，这些自动制造系统可以提供重要的信息，帮助管理者做出正确的商业决策。然而，由于自动制造系统具有高度灵活性，在系统运行过程中往往存在一些不稳定性。例如，错误的装配或者将工件放置在错误的存储器内。这类故障的发生不仅会降低系统生产率，带来经济上的影响，还有可能引起一系列不良的后果。因此，性能优化是自动制造系统中必须要考虑的问题。

对于一个自动制造系统，其加工生产时所使用的机器的数量和类型以及产品的质量都具有直接的经济效益。一旦资源没有被合理的分配利用，就会降低系统的生产效率，甚至会导致整个系统的瘫痪。因此，对于工程师或设计者而言，如何在有限资源的情况下寻找一种最优运行模式实现盈利最大化，或在保证一定生产力的前提下寻找一种最优运行模式使得成本最小化至关重要。时延Petri网模型作为一类有力的数学工具，广泛应用于自动制造系统的建模、分析和控制。它们可以用来分析系统的性能指标，解决实时的任务调度及资源最优化利用等问题。

本论文致力于研究时延Petri网建模的自动制造系统的性能优化问题。主要研究成果如下：

1. 对于一类确定性时延Petri网，时延带权标记图（TWMG）被广泛的应用于循环自动制造系统的建模与分析。对于单服务器语义下TWMG的标识优化问题，如何在保证系统产出的前提下寻找一个初始标识使得成本最小化是制造领域的重要问题之一。然而，现有的工作不能够提供有效解决问题的方法。在本文的工作中，我们充分利用TWMG的结构特性以及其活性相关知识给出了一个初始标识，并且提出了一种基于仿真的启发式算法，将托肯逐步加入部分库所来提高系统的产出。最后，结合以往工作的优点，我们引入了一种新的技术来进一步缩小成本。

2. 从实际应用的角度来说，服务器语义对应于一个操作过程中可使用的机器数目。如果资源充分多，在无穷服务器语义条件下，同一操作过程可以利用多个机器同时执行。而在单个服务器语义条件下，这一操作过程仅能利用一台机器

执行。作为单服务器语义的扩展，针对无穷服务器条件下TWMG的标识优化问题，本文提出了两种不同的启发式算法逐步逼近最优解。不仅如此，通过对每个变迁引入一个包含单个托肯的自环库所，这两种方法还可以应用到单服务器语义下TWMG的标识优化问题。

3. 同时，对于单服务器语义下的TWMG，如何在有限资源的情况下寻找一个初始标识使得系统产出最大化的循环时间优化问题，本文中也进行了相关研究。我们证明了在初始标识未知的情况下可以将TWMG转化为一系列等价的时延标记图(TMG)。因此对于一个TWMG求解最优初始标识的问题可以转换为对一系列TMG求解最优初始标识的问题。我们提出了一种基于混合整数线性规划（MILPP）的高效算法来解决循环时间优化问题。该方法的优点在于能够保证最优解。最后，进一步拓展至更一般的系统产出最大化和消耗资源最小化问题的研究。

4. 在此基础上，我们针对无穷服务器语义条件下TWMG的循环时间优化问题进行了研究。首先，我们证明了对于TWMG转化为一个等价TMG的周期性特性。并且，通过将TWMG的状态空间进行分类，得到了一系列的等价TMG。由此，我们提出了一种针对所有等价TMG求最优解的MILPP方法，从而可以得到该优化问题的最优解。但是由于状态空间分类的数目会随系统库所数目的增大而迅速增加，该方法面临较大的计算复杂度问题。因此我们提出了两种次优算法，避免了对所有分类进行穷举，从而大幅降低了求解MILPP时所需的计算量。

最后，在总结全文工作的基础上，我们对自动制造系统性能评估与性能优化的未来工作进行了展望。

关键词：离散事件系统， 时延Petri网， 带权标识图， 性能评估， 性能优化

ABSTRACT

In the last decades, there has been a constant increase in the awareness of company management about the importance of formal techniques in industrial settings to address problems related to monitoring and reliability, fault diagnosis, and optimal use of resources, during the management of plants. Of particular relevance in this setting are the so-called Automated Manufacturing Systems (AMSs), which are characterized by complex technological cycles that must adapt to changing demands. Modern AMSs are interconnected subsystems such as numerically controlled machines, assembly stations, automated guided vehicles, robots, conveyors and computer control systems. Manufacturers are using automated machines and controls to produce quality products faster and more efficiently. Meanwhile, these automated systems can provide critical information to help managers make good business decisions. However, due to the high flexibility of AMSs, failures such as a wrong assembly or a part put in a wrong buffer may happen during the operation of the system. Such failures may decrease the productivity of the system which has an economical consequence and can cause a series of disturbing issues. As a result, the *performance optimization* in AMSs are imperative.

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. Once the resources are not well assigned, the system may produce products with a low efficiency and even cause a deadlock. Therefore, the main problem for engineers or designers is to find an optimal mode of operations given a set of available resources or to find an optimal set of resources capable of meeting the required production constraints. As a powerful mathematical tool, timed Petri nets models have been extensively used to model, analyze, and control of AMSs. They can be used for performance analysis, tasks scheduling in real-time, and optimizing the use of resources.

This thesis focuses on the performance evaluation and performance optimization of automated manufacturing systems using timed Petri net models. The main results of this research are as follows.

1. We consider a class of deterministic timed Petri nets called *timed weighted marked graphs* (TWMGs), which are extensively used to model and analyze cyclic AMSs. The *marking optimization* of deterministic TWMGs under single server semantics plays an

important role in the manufacturing domain: it consists in finding an initial marking to minimize the cost of resources while the system's throughput is less than or equal to a given value. The existing results fail to provide practically effective and computationally efficient methods to analyze and solve this problem in such systems. We take the advantages of the net structural characteristics of a TWMG and utilize related knowledge of liveness of a TWMG to select a proper initial marking. Next, based on simulation a heuristic algorithm used to increase the system's throughput by iteratively adding tokens to some places is developed. Finally, a technique to reduce the cost of the obtained solution by taking the advantages of the previous works is proposed.

2. From a physical point of view, the server semantics can be interpreted as the number of servers that can be used to execute an operation. Under single server semantics, the same operation can only be executed once at a time, while the same operation can be executed as many times as the number of available servers under infinite server semantics. As an extension of single server semantics, this study proposes two efficient heuristic methods for the marking optimization problem of deterministic TWMGs under infinite server semantics. These proposed algorithms can provide a near optimal solution step by step and also apply for the marking optimization of deterministic TWMGs under k server semantics by adding to each transition a self-loop place with k token.

3. The *cycle time optimization* of deterministic TWMGs under single server semantics is originally studied in this research: it consists in finding an initial resource assignment to maximize the system's throughput while the cost of resources is less than or equal to a given value. We prove that a TWMG under single server semantics can be transformed into a series of equivalent *timed marked graphs* (TMGs) under the condition that the initial marking is not given. Hence the problem to determine an optimal initial marking for a TWMG can be converted to determining an optimal initial marking for a series of equivalent TMGs. A practically efficient algorithm is developed to solve the optimization problem based on solving a series of *mixed integer linear programming problems* (MILPPs), which can guarantee the convergence to the optimum. Finally, this approach is further extended to a generalized optimization problem which maximizes the system's throughput and minimizes the cost of the resources.

4. Based on previous results, the cycle time optimization of deterministic TWMGs under infinite server semantics is studied. We consider the transformation of a given TWMG

into an equivalent TMG under infinite server semantics and prove that this transformation is periodical with regard to the initial marking. This allow us to transform a TWMG into a finite family of equivalent TMGs, each one valid for a partition of set of initial markings. Then, we present an MILPP to solve the optimization problem that requires finding an optimal allocation for the equivalent TMG under the constraint that the initial marking belongs to a particular partition. However, this procedure has a high computational complexity due to the fact that the number of partitions can increase exponentially with the number of places. In order to reduce the computational complexity, two sub-optimal approaches are proposed without enumerating the entire partitions.

Finally, conclusions and future studies on performance evaluation and optimization for AMSs are prospected.

Keywords: Discrete event system, timed Petri net, weighted marked graph, performance evaluation, performance optimization

RÉSUMÉ

Au cours des dernières décennies, la complexité croissante des systèmes de production et de leur commande a rendu crucial le besoin d'utiliser les méthodes formelles pour faire face aux problèmes relatifs au contrôle, à la fiabilité, au diagnostic des fautes et à l'utilisation optimale des ressources dans les installations de production. Cela concerne en particulier les systèmes automatisés de production (SAP), caractérisés par des cycles technologiques complexes qui doivent s'adapter à des conditions changeantes. Les SAP modernes sont des sous-systèmes interconnectés tels que des machines à commande numérique, des stations d'assemblage, des véhicules guidés automatisés (AGV), des cellules robotisées, des convoyeurs et des systèmes de contrôle par ordinateur. Les fabricants utilisent des machines automatisées et des contrôleurs pour assurer des produits de qualité plus rapidement et plus efficacement. Aussi, ces systèmes automatisés peuvent fournir des informations essentielles pour aider les gestionnaires à prendre les bonnes décisions. Cependant, en raison de la grande flexibilité des SAP, des défaillances telles qu'un mauvais assemblage ou le dépôt d'une pièce dans un tampon inapproprié peuvent se produire lors du fonctionnement du système. De tels dysfonctionnements diminuent la productivité du système générant ainsi des pertes économiques et des effets perturbateurs sur le système. En conséquence, le problème de l'optimisation des performances des SAP est impératif.

La quantité de produits qui doivent être stockés ou déplacés, le nombre et le type de machines dans le système ont des conséquences économiques. Si les ressources ne sont pas bien affectées, la production risque d'être inefficace voire même complètement bloquée. Par conséquent, un problème principal pour les ingénieurs ou les concepteurs est de déterminer un mode d'exploitation optimal compte tenu des ressources disponibles ou de déterminer un ensemble optimal de ressources capable de satisfaire les contraintes de production requises. En tant qu'outil mathématique puissant, le formalisme des réseaux de Petri temporisés a été largement utilisé pour modéliser, analyser et contrôler les SAP. Il peut également être utile pour l'analyse des performances, la planification des taches en temps réel et l'optimisation de l'utilisation des ressources.

Cette thèse se focalise sur l'évaluation et l'optimisation des performances des systèmes de production automatisés via le modèle des réseaux de Petri temporisés.

Les principaux résultats obtenus dans cette recherche sont les suivants:

1. Les Graphes d'Evénements Temporisés généralisés (TWMG) forment une classe de réseaux de Petri temporisés largement utilisés pour modéliser et analyser les SAP cycliques (i.e. à production répétitive). Sous l'hypothèse que la politique de service soit celle du serveur unique (single serveur), le problème de l'optimisation du marquage des TWMG déterministes est important dans le domaine de la production. Il consiste à déterminer un marquage initial qui minimise le coût des ressources tout en assurant au système un débit donné. Les méthodes existantes pour analyser et résoudre ce problème ne sont efficaces ni en pratique ni en termes en terme de complexité algorithmique. En tirant avantage des caractéristiques structurelles des TWMG et des résultas connexes à la propriété de vivacité, nous sélectionnons un marquage initial approprié puis via une heuristique basée sur la simulation, on augmente le débit du système en ajoutant de manière itérative des jetons dans certaines places appropriées. Enfin, nous proposons une technique permettant de réduire le coût de la solution obtenue en exploitant les avantages des travaux précédents.

2. D'un point de vue physique, la sémantique de service peut être interprétée comme le nombre de serveurs pouvant être utilisés simultanément pour exécuter une opération (i.e. franchissement d'une transition). Sous la sémantique du serveur-unique, la même opération ne peut être exécutée qu'une seule fois (degré de franchissabilité égal à 1), alors que la même opération peut être exécutée autant de fois que l'on veut sous la sémantique de serveurs-infinis (infinite-servers). Dans notre étude, on propose deux méthodes heuristiques efficaces pour le problème d'optimisation du marquage des TWMGs déterministes sous la sémantique serveurs-infinis. Ces algorithmes proposés peuvent fournir une solution presque optimale en procédant étape par étape et s'appliquent également pour l'optimisation du marquage des TWMG déterministes sous la sémantique du serveur-unique en ajoutant à chaque transition une boucle de réentrance (self-loop) avec un jeton.

3. L'optimisation du temps de cycle des TWMGs déterministes sous la sémantique du serveur unique a été traitée dans cette recherche. Le problème consistant à trouver une affectation de ressources maximisant le débit du système tout en maintenant le coût des ressources inférieur ou égal à une valeur donnée. Nous prouvons qu'un TWMG sous une sémantique de serveur unique peut être ramené à une série de graphes d'événements temporisés (TMG) équivalents pour lesquels le marquage initial n'est pas donné. Par conséquent, le problème de détermination d'un marquage initial optimal pour

un TWMG peut être traduit au problème de détermination d'un marquage initial optimal pour une série de TMG équivalents. Un algorithme efficace en pratique est développé pour résoudre ce problème d'optimisation, il est basé sur la résolution d'une série de problèmes de programmation linéaire en nombres entiers mixtes (MILPP) garantissant ainsi la convergence à l'optimum. Enfin, cette approche est étendue au problème d'optimisation généralisé où l'on cherche à maximiser le débit du système et à minimiser le coût des ressources.

4. Sur la base des résultats précédents, l'optimisation du temps de cycle des TWMG déterministes sous la sémantique de serveurs-infinis est étudiée. Nous considérons la transformation d'un TWMG donné en un TMG équivalent sous la sémantique de serveurs-infinis et nous prouvons que cette transformation est périodique par rapport au marquage initial. Cela nous autorise de transformer un TWMG en une famille finie de TMG équivalents, chacun étant valable pour une partition de l'ensemble des marques initiaux. Ensuite, nous présentons un MILPP pour résoudre le problème d'optimisation qui exige la détermination d'une allocation optimale pour le TMG équivalent sous la contrainte que le marquage initial appartient à une partition particulière. Cependant, cette procédure présente une complexité en temps de calcul très de calcul élevée en raison du fait que le nombre de partitions croît exponentiellement avec le nombre de places. Afin de réduire cette complexité, deux approches sous-optimales ne nécessitant pas l'énumération entière des partitions sont proposées.

Mots-clés: Systèmes à événements discrets, Réseau de Petri temporisé, Graphes d'événements valués, Evaluation de performance, Optimisation de performance.

List of Figures

List of Tables

List of Symbols

N	A Petri net $N = (P, T, Pre, Post)$
N^{δ}	A timed Petri net $N = (P, T, Pre, Post, \delta)$
P	The set of places of a Petri net
T	The set of transitions of a Petri net
Pre	The input matrix of a Petri net
$Post$	The out matrix of a Petri net
C	The incidence matrix of a Petri net, $C = Post - Pre$
M	A marking of a Petri net
M_0	The initial marking of a Petri net
$\langle N, M_0 \rangle$	A Petri net system
$\mathcal{R}(N, M_0)$	The reachability set of a Petri net $\langle N, M_0 \rangle$
x	A minimal T-semiflow of a Petri net
y	A minimal P-semiflow of a Petri net
p	A place in a Petri net
t	A transition in a Petri net
$w(p)$	The weight of the unique input arc of place p
$v(p)$	The weight of the unique output arc of place p
gcd_p	The greatest common divisor of the integer $w(p)$ and $v(p)$
γ	An elementary circuit in a Petri net
Γ	The set of elementary circuits
y_γ	A minimal P-semiflow of circuit γ
λ_γ	The cost of the resources modeled by tokens in the support of y_γ
${}^\bullet p$	The set of the input transitions of place p
$p^\bullet$	The set of the output transitions of place p
${}^\bullet t$	The set of the input place of transition t
$t^\bullet$	The set of the output places of transition t
β	The throughput of a timed weighted marked graph system $\langle N, M_0 \rangle$
$\alpha_i(j)$	The enabling degree of transition t_i enabled at a marking M_j
$\chi(M_0)$	The cycle time of a timed weighted marked graph system $\langle N, M_0 \rangle$
$f(M)$	The cost of resources of marking M, $f(M) = y^T \cdot M$
M_D	A greatest dead marking, $M_D = (v(p_1) - 1, \ldots, v(p_n) - 1)^T$

b	An upper bound on the desired cycle time
R	An upper bound on the cost of resources
$\mathcal{M}_j$	A partition of the marking space of a TWMG net
ϕ	The transformation period of place p, $\phi = v(p) \cdot x_{out(p)}$
Φ	The number of marking partitions for a TWMG
Φ'	The reduced number of marking partitions for a TWMG
Φ''	The further reduced number of marking partitions for a single circuit
Φ'''	The further reduced number of marking partitions for a TWMG
Φ^4	The number of marking partitions for sub-optimal approaches
$\mathbb{N}$	The set of non-negative integers, $\mathbb{N} = \{0, 1, 2, \ldots\}$
$\mathbb{Z}$	The set of integers
$\mathbb{R}$	The set of real numbers

List of Abbreviations

AMS	automated manufacturing system
DES	discrete event system
FMS	flexible manufacturing system
MG	marked graph
WMG	weighted marked graph
TMG	(transition) timed marked graph
PTMG	place timed marked graph
TWMG	(transition) timed weighted marked graph
LPP	linear programming problem
ILPP	integer linear programming problem
MILPP	mixed integer linear programming problem

Content

Chapter 1 Introduction

The increasing global market competition has made manufacturing industries focus their attention on critical issues such as productivity and quality. Of particular relevance in this setting are the so-called Automated Manufacturing Systems (AMSs) whose importance is greatly recognized in both academic and industrial fields. Massive AMSs have been deployed in industrial companies to handle complex and hazardous operations instead of workers. As a result, both the quality and the efficiency of manufacturing system are improved, which make higher profits for the company. The performance analysis and control of such systems have became a hot topic in the field of academic and industrial.

An AMS consists of a set of workstations (each one capable of processing parts of different kind according to a prescribe sequence of operations) and interconnect subsystems that are composed by a large quantity of production lines, assembly stations, automated guided vehicles (AGVs), robots, conveyors, and other material-handling devices. Due to their high degree flexibility, it is necessary to reconfigure them on-line to find an optimal mode of operations given a set of available resources or to find an optimal set of resources capable of meeting the required production constraints. Disturbances, together with diagnosis and reconfigurations, constitute basic phenomena that we need to model for computing the real performance of an AMS [1]. The quantity of products which have to be stored or moved and the number and type of machines which operate the AMS have economical consequences. As a result, the main problem for designers is to find a trade-off between minimizing the cost of the resources and maximizing the system's throughput.

Petri nets [2] are a graph-based mathematical formalism for modeling and analyzing of discrete event systems (DESs) in a wide variety of applications [3]. As an efficient tool for describing and analyzing manufacturing systems, Petri nets have found their extensive applications to the supervisory control [4–20], analysis [21–29], deadlock prevention of AMSs [30–40], and fault diagnosis [41–54]. However, in real manufacturing systems, activities do not take place instantaneously. Every activity in a manufacturing system has a time duration which is different from zero. As a result, three types of Petri nets with timing information are proposed in the literature: Petri nets with interpretation of time in the transitions [23], Petri nets with interpretation of time in the places [55], and Petri nets with time-dependent arcs [29]. As an extension formalism of Petri nets, time Petri nets are a discrete event models that associate with the time instants in which events occur and find wide applications in manufacturing systems and embedded systems [56–65]. They can be used for performance analysis of a system, i.e., speeds of a process, tasks scheduling, optimizing the use of resources, and so on.

1.1 Performance Estimation

Performance estimation of batch processes or high throughput manufacturing systems poses difficult problems since their representation deals with discrete models. Based on time Petri nets, researchers have developed many policies to study the performance estimation problems in AMSs [1, 23, 66–70]. Generally, there mainly exit two analysis techniques to deal with the performance evaluation in AMSs: simulation approach [66, 71–76], analytical approach based on linear programming problem (LPP) technique [67–69, 77–79] and based on *tropical algebra* like (max,+) or (min,+) [80–86]. The former one is usually straightforward and effective to study the evolution and dynamic behaviors of the system and can provide an exact value of performance. However, due to the *state explosion* problem, this method cannot be applied to large scaled manufacturing systems, where the number of states grows exponentially with respect to the size of the system. For the analytical approaches based on both LPP technique and tropical algebra, structural properties of Petri nets are fully utilized and the state explosion problems are avoided. As a result, these approaches can reduce the computational cost significantly and provide bounds or an exact value of performance. However, it is difficult to evaluate the accuracy of the obtained bound with respect to the real system performance.

Simulation of time Petri nets has been demonstrated to be useful for analyzing transient and permanent behavior of DESs in performance evaluation. Based on this technique, an efficient algorithm for the execution of time Petri net is proposed in [72]. Several friendly user interface tools are developed for analysis of time Petri nets [87–89]. The study in [90] deals with the analysis of timed discrete, continuous and hybrid Petri nets. By contrast to time Petri net where each transition or place is associated with a time interval, in timed Petri net each transition or place is associated with a time duration, i.e., a single value. As a conclusion, most of the performance estimation studies based on simulation focus on the analysis of steady states and the average firing rate of transitions.

Analytical method-based performance estimation policy is a typical application of structure analysis techniques of Petri nets. Performance bounds are evaluated which can avoid the necessity of enumerating the whole state space in a Petri net. In [91, 92], the authors present an approach for the analysis of dynamic behavior and performance based on the computation of a *state class graph*. In the work of Ramchandani [23] manufacturing systems are modeled by deterministic timed marked grpahs (TMGs) and an analytical method based on solving LPP is firstly proposed which provides a bound of the performance. Based on the same techniques, properties and performance bounds of timed and stochastic marked graphs are studied in [67, 68, 78] and these bounds depend on both the initial marking and the average values of the delays of transitions [69]. Nevertheless, these approaches fail to provide a bound that is close to the real value. To fulfill this goal, Rodriguez *et al.* [93] propose an iterative strategy to obtain an upper bound which closer to the real performance than previous works. In each iteration step, the bottleneck circuit is searched by solving an LPP and the parts which may constrain the current bottleneck circuit are added to calculate a new upper bound.

However, TMGs cannot model important features that may be present in manufacturing systems such as the processing of parts in batches whose size may change during different processing steps. For this reason, a more general model called *timed weight marked graphs* (TWMGs) is studied in [27]: this model is characterized by weighted arcs. The studies of performance estimation for manufacturing systems modelled by timed weighted marked graphs (TWMGs) are discussed in several works. By transforming a TWMG into an *equivalent* TMG, Munier [94] proposes a pseudo-polynomial algorithm to compute the performance of a TWMG under *single server* semantics. Nakamura and Silva [95] discuss the same problem under the *infinite server* semantics and a similar transformation technique is developed. However, the disadvantage of the approaches in [94, 95] is that the transformation can lead to a model of significant size.

Tropical algebras have been broadly used to describe the behavior and analyse the performance. The behaviors and performance of TMGs are described by recurrent linear equations in (min,+) algebra [80, 81, 96] or in (max, +) algebra [85, 86]. However, the weights on the arcs of a TWMG lead to non-linear models in tropical algebra. Thus, a linearization method is proposed in [97] when each elementary circuit contains at least one *unitary* transition (i.e., a transition for which its corresponding elementary T-semiflow component is equal to one). This method increases the number of transitions. Inspired by this work, some linearization methods without increasing the number of transitions are proposed in [98–104]. The obtained (min,+) linear model allows to evaluate the performance of TWMGs.

Other works provide bounds for embedded data flow systems by using synchronous data flow graphs are investigated in [105–108]. Performance estimation of such systems is a critical step to verify throughput requirements of concurrent real-time applications. These studies requires transformation to another kind of data flow graph, which can lead to a model of significant size with respect to the original graph. In the work of Ghamarian *et al.* [109] a method without transformation is proposed. This method generates and analyzes the dynamic state space of the graphs by executing it. In [110] a class of closed queueing networks is studied and performance upper and lower bounds are estimated by using LPPs.

1.2 Performance Optimization

Performance optimization of manufacturing systems in time Petri nets have been extensively studied in the literature. The optimization problem is solved by heuristic algorithms such as genetic algorithms, simulated annealing and threshold accepting, and analytical approaches based on integer linear programming problem (ILPP).

Based on time Petri nets, researchers have provided many policies to deal with the performance optimization in manufacturing systems. Rodriguez *et al.* [93] deal with the resources optimization in process Petri nets and presents a heuristic strategy to gauge in the best possible way the number of resources needed so that the overall system throughput is maximized. In order to avoid the state explosion problem, the proposed techniques take the full advantage of the structural property. The considered process Petri nets are assumed to be live by pre-assigning tokens to resource places and deadlock-free problem is not addressed. Stochastic approximation algorithms are provided to solve the performance optimization problem of stochastic Petri nets [111, 112]. Chen *et al.* [113] develop a new model, called batch deterministic and stochastic PNs, to model batch features in supply chains and study the performance optimization problem.

Due to the competition for limited number of resources among concurrently executed production lines, it may result in a deadlock situation. Thus, the scheduling problems in manufacturing systems where resources are shared by multiple processes are very important. Abdallah *et al.* [114] present a deadlock-free scheduling algorithm for a class of systems called systems of sequential systems with shared resources in timed Petri nets. The algorithm generates a partial reachability graph to find the optimal or near optimal deadlock-free schedule. Wu and Zhou [115] solve the real-time deadlock-free scheduling problem for semiconductor track systems based on colored timed Petri nets in a hierarchical way. A deadlock avoidance policy is developed for the system as a lower-layer controller and then heuristic rules are proposed to schedule the system in real-time. Based on genetic algorithm, Xing *et al.* [116] develop a deadlock-free genetic scheduling algorithm to avoid the deadlock situation and minimize the makespan. By using the one-step look-ahead method in the optimal deadlock control policy, the feasibility of a chromosome is checked and infeasible chromosomes are amended into feasible ones, which can be easily decoded into a feasible deadlock-free schedule.

By contrast to the aforementioned works, several works are dedicated to the performance optimization for TMGs and TWMGs which are conflict free nets, i.e., there exist no shared resources [73, 74, 117–120]. These problems of Petri nets are very important for the design of many exemplified discrete event dynamic systems in the real world. In particular, two classical performance optimization problems are commonly considered in the literature: marking optimization problem (also called the minimum cost initial distributed state problem) and cycle time optimization problem (also called the maximum throughput initial state assignment problem). The marking optimization problem aims to find a proper schedule which minimizes the cost of resources under the constraint that the system's throughput should not smaller than a given value, while the cycle time optimization problem aims to find a proper schedule which maximizes the system's throughput under the constraint that the cost of resources should not exceed a given bound. Meanwhile, both single server semantics and infinite server semantics are investigated for the performance optimization problems. From a physical point of view, the server semantics can be interpreted as the number of times that an operation can be executed concurrently. Under single server semantics, the same operation can only be executed once at a time, while the same operation can be executed as many times as the number of available servers under infinite server semantics.

Laftit *et al.* study the marking optimization problem for TMGs under infinite server semantics [63, 121] and provides a heuristic algorithm and an exact algorithm to find a near optimal solution. Gaubert addresses the same problem by using *min-max algebra* [122, 123]. Giua *et al.* deal with the cycle time optimization problem for TMGs under infinite server semantics [120] and proposes three different approaches to find an optimal solution. However, in the literature, few works deal with the optimization problem of TWMGs. Benazouz *et al*. [58] develop an algorithm to minimize the overall buffer capacities with throughput constraint for TWMGs. Sauer proposes a heuristic solution based on an iterative process to solve the marking optimization problem of TWMGs under single server semantics [73]. Nevertheless, the presented solutions are heuristic and the optimality is not ensured. Thus, the problem of finding an optimal solution for marking optimization and cycle time optimization of TWMGs is still open.

1.3 Thesis Organization

This thesis focuses on the performance optimization of AMSs in the DES model of timed Petri nets.

Chapter 2 provides the basics of timed Petri nets and some notations used in the rest of the thesis. In particular, two subclasses of timed Petri nets (TMGs and TWMGs) as well as their dynamic evolutions and the concepts of cycle time are introduced.

In Chapter 3, we focus on the marking optimization of deterministic TWMGs under single server semantics. The problem consists in finding an initial marking to minimize the cost of resources while the system's throughput is less than or equal to a given value. The existing results fail to provide practically effective and computationally efficient methods to analyze and solve the problem in such systems. We take the advantages of the net structure characteristics of a TWMG and utilize related knowledge of liveness of a TWMG to select a proper initial marking. Next, based on simulation a heuristic algorithm used to increase the system's throughput by iteratively adding tokens to some places is developed. Finally, a technique to reduce the cost of the obtained solution by taking the advantages of the previous works is proposed. Numerical simulation studies show that the proposed method requires less iteration steps and thus is much faster than the previous approach.

As an extension problem of the one in Chapter 3, Chapter 4 investigates the marking optimization of deterministic TWMGs under infinite server semantics. From a physical point of view, the server semantics can be interpreted as the number of servers that can be used to execute an operation. Under single server semantics, the same operation can only be executed once at a time, while the same operation can be executed as many times as the number of available servers under infinite server semantics. Two efficient heuristic methods are proposed to obtain a near optimal solution step by step. The proposed methods also apply for the problem in Chapter 3 by adding to each transition a self-loop place with one token.

In Chapter 5, the cycle time optimization of deterministic TWMGs under single server semantics is originally studied. We aim to find an initial marking which maximizes the system's throughput, while the cost of resources is less than or equal to a given value. We prove that a TWMG under single server semantics can be transformed into a series of equivalent TMGs under the condition that the initial marking is not given. Hence the problem to determine an optimal initial marking for a TWMG can be converted to determine an optimal initial marking for a series of equivalent TMGs. A practically efficient algorithm is developed to deal with the optimization problem based on solving a series of MILPPs. Finally this approach is further extended to a generalized optimization problem which maximizes the system's throughput and minimizes the cost of the resources.

Based on previous results, the cycle time optimization of deterministic TWMGs under infinite server semantics is studied in Chapter 6. We consider the transformation of a given TWMG into an equivalent TMG under infinite server semantics and prove that this transformation is periodical with regard to the initial marking. This allow us to transform a TWMG into a finite family of equivalent TMGs, each one valid for a partition of set of initial markings. Then, we present an MILPP to solve the optimization problem that requires finding an optimal allocation for the equivalent TMG under the constraint that the initial marking belongs to a particular partition. However, this procedure has a high computational complexity due to the fact that the number of partitions can increase exponentially with the number of places. In order to reduce the computational complexity, two sub-optimal approaches are proposed without enumerating the entire partitions.

Finally, Chapter 7 concludes this dissertation and provides some future directions of the work.

Chapter 2 Preliminary

The concept of Petri nets originates from Carl Adam Petri's doctoral dissertation. As an extension formalism of Petri nets, timed Petri nets are firstly introduced in Chander Ramchandani's doctoral dissertation. In this chapter, the basic concepts, definitions, dynamic behaviors of timed Petri nets used in this thesis are given.

2.1 Petri Nets

Definition 2.1. A Petri net is a four-tuple $N = (P, T, Pre, Post)$, where P is a set of n *places*; T is a set of m *transitions*; $Pre : P \times T \to \mathbb{N}$ and $Post : P \times T \to \mathbb{N}$ are the *pre-* and *post-incidence* functions that specify the arcs in the net; $C = Post - Pre$ is the *incidence matrix*. □

Definition 2.2. A Petri net $N = (P, T, Pre, Post)$ is said to be *ordinary* when all its arc weights are unitary. □

Definition 2.3. A *marked graph* (MG) is an ordinary Petri net such that each place has only one input transition and one output transition. A *weighted marked graph* (WMG) is a net such that each place has only one input transition and one output transition but may not be ordinary, i.e., the weight associated with each arc is a positive integer number. □

Definition 2.4. A *marking* $M : P \to \mathbb{N}$ of a Petri net is a mapping that assigns a non-negative integer of tokens to each place; $M(p)$ denotes the marking of place p. A *Petri net system* $\langle N, M_0 \rangle$ is a net N with an *initial marking* M_0. □

Graphically, places and transitions are denoted by circles and bars, respectively. Each directed arc is labeled by positive integers to represent their weights. An arc without a label indicates that its weight is unitary. Tokens in a place is denoted by black dots or a positive integer representing their quantity. From the physical point of view, a place represents an operation or a state of a resource while a transition represents the start or end of an operation. A token in a place means the fulfilment of a condition or the availability of a resource.

Example 2.1. Consider the Petri net system $\langle N, M_0 \rangle$ in Fig. 2.1. In this net, $P = \{p_1, p_2, p_3,$

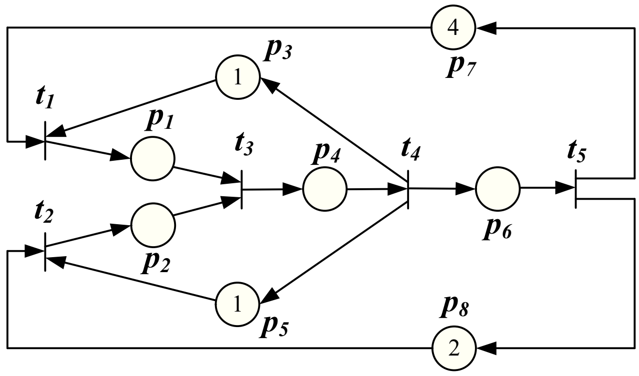

Fig. 2.1 A marked graph system $\langle N, M_0 \rangle$.

$p_4, p_5, p_6, p_7, p_8\}$, $T = \{t_1, t_2, t_3, t_4, t_5\}$,

$$Pre = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \qquad Post = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Its incidence matrix is:

$$C = Post - Pre = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 1 \end{bmatrix}.$$

The initial marking of this net is

$$M_0 = (0, 0, 1, 0, 1, 0, 4, 2)^T.$$

It is easy to verify that the net is ordinary. ◇

Definition 2.5. A vector $x = (x_1, x_2, \ldots, x_m)^T \in \mathbb{N}^{|T|}$ is called a *T-semiflow* iff $x \neq \mathbf{0}$ and $C \cdot x = \mathbf{0}$. A vector $y = (y_1, y_2, \ldots, y_n)^T \in \mathbb{N}^{|P|}$ is called a *P-semiflow* iff $y \neq \mathbf{0}$ and $y^T \cdot C = \mathbf{0}$. The supports of a T-semiflow and a P-semiflow are defined by $\|x\|$=$\{t_i \in T | x_i > 0\}$ and $\|y\|$=$\{p_i \in P | y_i > 0\}$, respectively. A *minimal* T-semiflow [1] (resp., P-semiflow) is

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

a T-semiflow $\|x\|$ (resp., P-semiflow $\|y\|$) whose support is not a superset of the support of any other T-semiflow (resp., P-semiflow), and whose components are mutually prime. □

Definition 2.6. Given a net $N = (P, T, Pre, Post)$ and a marking M, a transition t is enabled at M if $M \geq Pre(\cdot, t)$ and is denoted as $M[t\rangle$. An enabled transition t may fire yielding a new marking M' with

$$M' = M + C(\cdot, t), \tag{2-1}$$

where $Pre(\cdot, t)$ (resp., $C(\cdot, t)$) denotes the column of the matrix Pre (resp., C) associated with transition t. Marking M'' is said to be *reachable* from M if there exists a sequence of transitions $\sigma = t_0 t_1 \ldots t_n$ and markings $M_1, M_2, \ldots$, and M_n such that $M[t_0\rangle M_1[t_1\rangle M_2 \ldots M_n[t_n\rangle M''$ holds. The set of markings reachable from M_0 in $\langle N, M_0\rangle$ is called the *reachability set* of the Petri net system $\langle N, M_0\rangle$ and denoted as $\mathcal{R}(N, M_0)$. □

Definition 2.7. Given a Petri net system $\langle N, M_0\rangle$, $t \in T$ is live under M_0 iff $\forall M \in \mathcal{R}(N, M_0)$, $\exists M' \in \mathcal{R}(N, M_0)$, $M'[t\rangle$. $\langle N, M_0\rangle$ is live iff $\forall t \in T$ is live under M_0. □

Example 2.2. In the net $\langle N, M_0\rangle$ shown in Fig. 2.1, there are four minimal P-semiflows: $y_1 = (1,0,1,1,0,0,0,0)^T$, $y_2 = (1,0,0,1,0,1,1,0)^T$, $y_3 = (0,1,0,1,1,0,0,0)^T$, and $y_4 = (0,1,0,1,0,1,0,1)^T$, since $\forall i \in \{1,2,3,4\}$, $y_i^T \cdot C = \mathbf{0}$, and a unique minimal T-semiflow $x_1 = (1,1,1,1,1)^T$ since $C \cdot x_1 = \mathbf{0}$.

Transitions t_1 and t_2 are enabled at the initial marking M_0. By firing t_2 at $M_0 = (0,0,1,0,1,0,4,2)^T$, we obtain a new marking $M_1 = (0,1,1,0,0,0,4,1)^T$, i.e., $M_0[t_2\rangle M_1$, which can be verified by Eq. (2-1) as follows:

$$M_0 + C(\cdot, t_2) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 4 \\ 2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 4 \\ 1 \end{bmatrix} = M_1$$

From M_0 by firing a sequence $\sigma = t_2 t_1 t_3$, it yields a marking $M_3 = (0,0,0,1,0,0,3,1)^T$ which is denoted as $M_0[t_2\rangle M_1[t_1\rangle M_2[t_3\rangle M_3$. The net is live since all transitions are live. ◇

minimal semiflow.

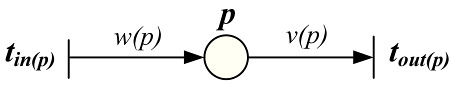

Fig. 2.2 A place p with a single input transition $t_{in(p)}$ and a single output transition $t_{out(p)}$.

Definition 2.8. Given a net $N = (P, T, Pre, Post)$, the *set of input places* (resp., *set of output places*) for a transition $t \in T$ is defined as ${}^\bullet t = \{p \in P \mid Pre(p,t) > 0\}$ (resp., $t^\bullet = \{p \in P \mid Post(p,t) > 0\}$). For a place $p \in P$, the *set of its input transitions* (resp., *set of its output transitions*) is defined as ${}^\bullet p = \{t \in T \mid Post(p,t) > 0\}$ (resp., $p^\bullet = \{t \in T \mid Pre(p,t) > 0\}$). □

Example 2.3. Consider a place p in Fig. 2.2 with single input transition $t_{in(p)}$ and single output transition $t_{out(p)}$. Let $w(p)$ and $v(p)$ be the weights of its input arc and output arc, i.e., $w(p) = Post(p,t)$, $v(p) = Pre(p,t)$. We denote the greatest common divisor of $w(p)$ and $v(p)$ by gcd_p. ◇

Definition 2.9. A Petri net is said to be *strongly connected* if there exists a directed path from any node in $P \cup T$ to every other node. An *elementary circuit* of a Petri net is a directed path that goes from one node back to the same node without passing twice on the same node and is denoted as γ. The set of elementary circuits is denoted as Γ. □

In a strongly connected Petri net, it is easy to show that each node belongs to an elementary circuit, and thus the name cyclic nets is also used to denote this class.

2.2 Weighted Marked Graphs

Definition 2.10. A *marked graph* (MG) is an ordinary Petri net such that each place has only one input transition and one output transition. □

Definition 2.11. A *weighted marked graph* (WMG) is a net such that each place has only one input transition and one output transition but may not be ordinary, i.e., the weight associated with each arc is a positive integer number. □

Definition 2.12. An elementary circuit γ of a WMG is said to be *neutral* if the following condition holds:

$$\prod_{p\in\gamma} \frac{v(p)}{w(p)} = 1. \quad \square$$

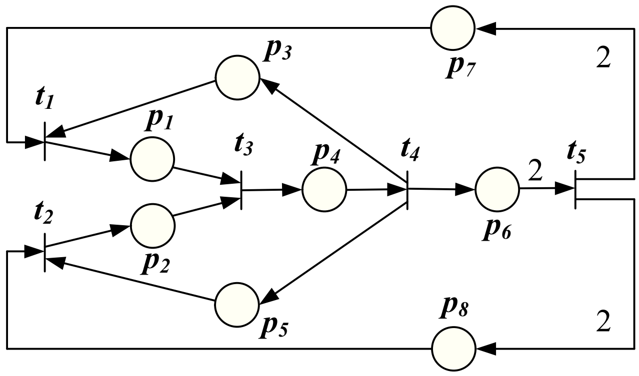

Fig. 2.3 A weighted marked graph $N = (P, T, Pre, Post)$.

In other words, in a neutral circuit the product of the weights of all pre-arcs is equal to that 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.

Definition 2.13. A WMG is *neutral* iff all its elementary circuits are neutral. □

Proposition 2.1. A strongly connected and neutral WMG 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.

Property 2.1. A strongly connected and neutral WMG has a *unique* minimal T-semiflow $\boldsymbol{x}$ which contains all transitions in its support. □

In the rest of the thesis, we will consider strongly connected and neutral WMGs.

Example 2.4. Consider the WMG net $N = (P, T, Pre, Post)$ in Fig. 2.3 which is strongly connected and consists of four elementary circuits $\gamma_1 = p_1t_3p_4t_4p_6t_5p_7t_1$, $\gamma_2 = p_2t_3p_4t_4p_6t_5p_8t_2$, $\gamma_3 = p_1t_3p_4t_4p_3t_1$, and $\gamma_4 = p_2t_3p_4t_4p_5t_2$. For circuit γ_1, the product of the weights of all post-arcs is equal to

$$w(p_1) \cdot w(p_4) \cdot w(p_6) \cdot w(p_7) = 2,$$

while the product of the weights of all pre-arcs is equal to

$$v(p_1) \cdot v(p_4) \cdot v(p_6) \cdot v(p_7) = 2.$$

Thus, circuit γ_1 is neutral. Similarly, circuits γ_2, γ_3, and γ_4 are also neutral. The WMG is neutral since all its elementary circuits are neutral. As a result, it has a unique T-semiflow

$x = (2, 2, 2, 2, 1)^T$. There are four P-semiflows corresponding to the four elementary circuits:

$$\begin{cases} \gamma_1 : \ y_1 = (1, 0, 0, 1, 0, 1, 1, 0)^T, \\ \gamma_2 : \ y_2 = (0, 1, 0, 1, 0, 1, 0, 1)^T, \\ \gamma_3 : \ y_3 = (1, 0, 1, 1, 0, 0, 0, 0)^T, \\ \gamma_4 : \ y_4 = (0, 1, 0, 1, 1, 0, 0, 0)^T. \end{cases}$$

◇

2.3 Timed Weighted Marked Graphs

The timing structure of a net can be *deterministic* when the delays are known a priori, or *stochastic* when the delays are random variables. There mainly exist two ways of introducing the timing structure in Petri net models, i.e., associating the timing structure with either transitions or places[2]. It has been proved that the two models are equivalent and can be easily transformed with each other [55].

Definition 2.14. A *deterministic transition timed* Petri net is a pair $N^\delta = (N, \delta)$, where $N = (P, T, Pre, Post)$ is a standard Petri net, and $\delta : T \to \mathbb{N}$, called delay time[3], assigns a non-negative integer fixed duration $\delta(t_j)$ to each transition t_j. In terms of a *deterministic place timed* Petri net, each place p_i is assigned a non-negative integer number $\delta(p_i)$ which represents the sojourn time that a token must spend in place p_i before it becomes available for its output transition. □

In the rest of this thesis, we will consider *weighted marked graphs* that are *deterministic transition timed* and call them TWMGs. When a transition t_i becomes enabled, it cannot fire before the time $\delta(t_i)$ has elapsed. Under the *As Soon As Possible* (ASAP) execution policy, a transition t_i will fire *exactly* after t_i is enabled for a time $\delta(t_i)$. The logical enabling condition for transitions must hold consecutively, i.e., transitions have only memory of the current enabling.

Definition 2.15. The *enabling degree* of transition t_i enabled at a marking M_j, denoted by $\alpha_i(j)$, is the biggest integer number k such that

$$M_j \geq k \cdot Pre(\cdot, t_i).$$ □

[2]Few works discuss the timing structure associated with arcs [29].

[3]If the delays are rational numbers everything in this thesis works the same by changing the time unit.

Another fundamental notion that should to be specified when defining a deterministic timed Petri net is the so-called *server semantics*.

- *Single server semantics*: each transition represents an operation that can be executed by a single operation unit.
- *Infinite server semantics*: each transition represents an operation that can be executed by an infinite number of operation units that work in parallel.
- k*-server semantics*: each transition represents an operation that can be executed by a finite number k of operation units.

In the case of timed Petri nets under single server semantics, services in a transition are provided sequentially. On the contrary, under infinite server semantics the number of concurrent servers is equal to the enabling degree of the transition. Note that infinite server semantics is more general than single server (or in general k-server) semantics. In fact, single (resp., k) server semantics can be simulated by infinite server semantics adding to each transition a self-loop place with one (resp., k) tokens.

A *clock* o_i associated with an enabled transition t_i at marking M represents the residual time to fire t_i. The server semantics specifies as many clocks are associated with an enabled transition:

- *Single server semantics*: one clock.
- *Infinite server semantics*: as many clocks as its enabling degree.
- k*-server semantics*: a number of clocks equal to min (k, $\alpha_i(j)$).

Under infinite server semantics, at each *time instant* τ_j the number of clocks o_i associated with a transition t_i is equal to its current enabling degree, i.e., $o_i = \{o_{i,1}, \ldots, o_{i,\alpha_i(j)}\}$; this number changes with the enabling degree, thus it can change each time the net evolving from one marking to another one, namely, each time a transition fires. If transition t_i is not enabled at marking M_j, its clock is an *empty set*. Assume that

$$o_i^* = \min\{o_{i,1}, \ldots, o_{i,\alpha_i(j)}\}$$

and let

$$o^* = \min_{i=1,\ldots,m} \{o_i^*\}$$

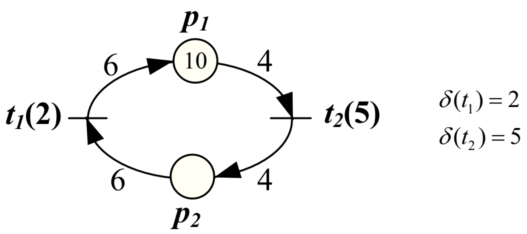

Fig. 2.4 A timed weighted marked graph system $\langle N^{\delta}, M_0 \rangle$.

be the minimum among the values of the clocks o_i^*. At the time instant

$$\tau_{j+1} = \tau_j + o^*,$$

transitions whose clocks are equal to o^* fire, yielding a new marking as in Eq. (2-1).

If the minimal value of the clock o_i^* at marking M_j holds for more than one clock, as an example k, in the set $\{o_{i,1}, \ldots, o_{i,\alpha_i(j)}\}$, implying that if the transition will be the next one to fire, it will fire k times simultaneously under infinite server semantics.

Definition 2.16. The *state* of a timed Petri net $[M; O]$ is defined not only by the marking M, as for Petri nets, but also by the clocks $O = (o_1, \ldots, o_n)$ associated with transitions. □

Example 2.5. Consider a TWMG system $\langle N^{\delta}, M_0 \rangle$ shown in Fig. 2.4. The net structure is $N = (P, T, Pre, Post)$ and the initial marking is $M_0 = (10, 0)^T$. The delay times are $\delta(t_1) = 2$ and $\delta(t_2) = 5$.

Under single server semantics, t_2 is enabled once at marking M_0 at initial time instant $\tau_0 = 0$ and its clock is $O_0 = (o_1, o_2) = (\emptyset, 5)$. After five time instants, i.e., at τ_1=$\tau_0 + \delta(t_2)$, transition t_2 will fires yielding a new marking $M_1 = (6, 4)^T$ with clock $O_1 = (\emptyset, 5)$.

Under infinite server semantics, the enabling degree of transition t_2 at marking M_0 is $\alpha_2(0) = 2$, i.e., it has two active clocks $o_{2,1}$ and $o_{2,2}$. The clock of marking M_0 is $O_0 = (\emptyset, \{5, 5\})$. After five time instants, i.e., at τ_1=$\tau_0 + \delta(t_2)$, transition t_2 fires twice yielding a new marking $M_1 = (2, 8)^T$ with clock $O_1 = (2, \emptyset)$. ◇

2.4 Cycle Time of Timed Weighted Marked Graphs

Definition 2.17. The *cycle time* $\chi(M)$ of a TWMG system $\langle N^{\delta}, M \rangle$ is the average time to fire once the minimal T-semiflow under the ASAP operational model. We denote the cycle time of an elementary circuit γ by $\chi_{\gamma}(M)$. □

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.

Definition 2.18. The *cycle time of transition* t_i of a TWMG is the average time between two consecutive firings of t_i, which is equal to

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

Definition 2.19. Let $t_i \in T$ be an arbitrary transition of a TWMG with the minimal T-semiflow x. The *cycle time* of the TWMG is equal to

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

The value of the cycle time does not depend on the considered transition. It is proved that the ASAP execution of a live and strongly connected timed marked graphs (TMG) with integer delays is ultimately repetitive following a periodical pattern of period Ψ [124, 125]. In the case of TWMGs, the ASAP execution is also ultimately periodic. Fig. 2.5 shows the evolution of an arbitrary live and strongly connected TWMG, where M_0 is the initial marking and the arrows correspond to ASAP execution steps. From M_u, the system will enter a cycle whose period is

$$\Psi = \tau_u - \tau_q,$$

and the number of firings of transition $t_i \in T$ within the steady period is f_i. This value is not identical for each transition but the proportion is equal to the minimal T-semiflow x. Thus, the cycle time of the TWMG system $\langle N^\delta, M_0 \rangle$ is equal to

$$x_i \cdot \frac{\tau_u - \tau_q}{f_i}. \tag{2-2}$$

Example 2.6. Consider the TWMG $N^\delta = (P, T, Pre, Post, \delta)$ whose net structure, initial marking, and timing structure are shown in Fig. 2.4, where $x = (2,3)^T$, $M_0 = (10,0)^T$, $\delta(t_1) = 2$, and $\delta(t_2) = 5$.

The evolution of the TWMG under single server semantics is presented in Fig. 2.6 (a). States $[M_1;\ O_1]$ and $[M_6;\ O_6]$ are the same, implying that from state $[M_6;\ O_6]$ the system

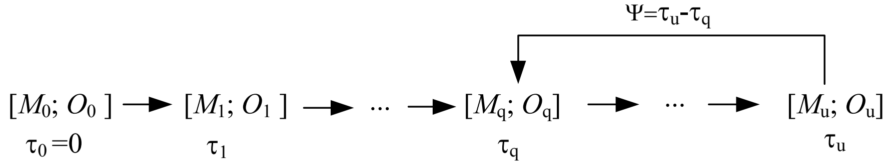

Fig. 2.5 Evolution of an arbitrary live and strongly connected TWMG.

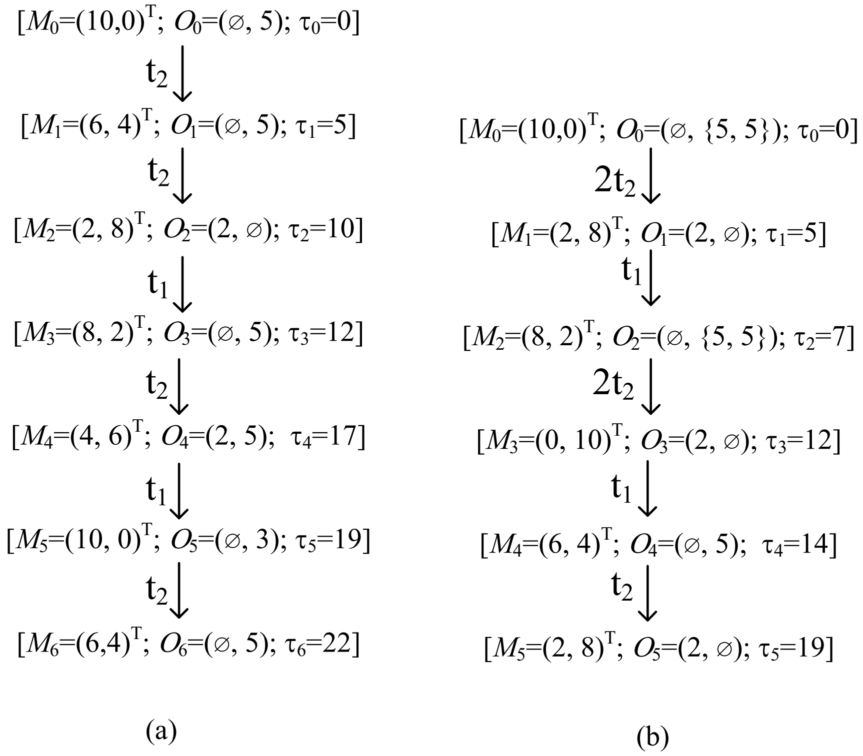

Fig. 2.6 (a) Evolution of the TWMG model for Example 2.6 under single server semantics; (b)Evolution of the TWMG model for Example 2.6 under infinite server semantics.

will enter a cycle which includes five states and the repetitive firing sequence is $t_2t_1t_2t_1t_2$. The period of the cycle is $\Psi = \tau_6 - \tau_1 = 17$ and the number of firings of transition t_1 (resp., t_2) within the steady period is 2 (resp., 3). Thus, the cycle time of the TWMG system $\langle N^\delta, M_0\rangle$ under single server semantics is equal to 17 by solving Eq. (2-2).

The evolution of the TWMG under infinite server semantics is shown in Fig. 2.6 (b). States $[M_1;\ O_1]$ and $[M_5;\ O_5]$ are the same, implying that from state $[M_5;\ O_5]$ the system will enter a cycle which includes four states and the repetitive firing sequence is $t_12t_2t_1t_2$. Note that the enabling degree of transition t_2 at marking M_2 is equal to two, which means that t_2 will fire twice simultaneously yielding marking M_3. The period of the cycle is $\Psi = \tau_5 - \tau_1 = 14$ and the number of firings of transition t_1 (resp., t_2) within the steady period is 2 (resp., 3). Thus, the cycle time of the TWMG system $\langle N^\delta, M_0\rangle$ under infinite server semantics is equal to 14 by solving Eq. (2-2). $\diamond$

The cycle time of a TWMG depends on the cycle time of its circuits. Let $\chi^\star(M) = \max_{\gamma\in\Gamma} \chi_\gamma(M)$ be a *critical time*. Any $\gamma \in \Gamma$ such that $\chi_\gamma(M) = \chi^\star(M)$ is a *critical circuit* that is denoted as $\gamma^\star$. It is well known that the cycle time of a TMG is equal to the critical time, i.e.,

$$\chi(M) = \chi^\star(M). \tag{2-3}$$

However, this result does not apply to a TWMG as we will show in this thesis.

Chapter 3 Marking Optimization of TWMGs Under Single Server Semantics

This chapter copes with the marking optimization problem (also called the minimum cost initial distributed state problem) of deterministic TWMG which consists in finding an initial marking to minimize the weighted sum of tokens in places while the cycle time is less than or equal to a given value. In addition, the server semantics considered in this chapter is single server semantics. We propose an iterative heuristic algorithm to solve the marking optimization problem. At each step, we select places from some circuits to which useful tokens are added until the cycle time is less than or equal to the desired value. Numerical simulation studies show that the proposed method requires less iteration steps and thus is much faster than the approach in [73].

3.1 Introduction

Petri nets have found their extensive applications to the supervisory control [4, 5, 19, 20], analysis [22, 24, 26–29], deadlock prevention of AMSs [31–33, 35, 36, 40], and *fault diagnosis* [41, 44, 45, 48, 49]. However, in real manufacturing systems, activities do not take place instantaneously. Every activity in a manufacturing system has a time duration which is different from zero. Timed Petri nets are well known as efficient tools for modeling discrete event systems and representing their dynamic behaviors. They can be used for performance analysis of a system, i.e., speeds of a process, tasks scheduling, optimizing the use of resources, and so on.

Timed weighted marked graphs and timed marked graphs (TMGs) are two important subclasses of timed Petri net that 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 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 problems for designers is to find a optimal set of resources capable of meeting the required production constraints. This problem is also well known as *marking optimization problem*.

For TMGs, marking optimization problem has been extensively studied in the past decades. Panayiotou and Cassandras [126] develop two incremental optimization algorithms to maximize a given performance index by assigning a set of resources step by step. Laftit *et al.* study the marking optimization problem for TMGs under infinite server semantics [63, 121] which provides a heuristic algorithm and an exact algorithm to find a near optimal solution. Gaubert addresses the same problem by using *min-max algebra* [122, 123]. Proth *et al.* propose a branch and bound method to obtain a near optimal solution [62]. However, in the literature, few studies are found to consider the marking optimization problem for TWMGs. Sauer [73] deals with the problem of finding an initial marking to minimize the weighted sum of tokens in places while the cycle time is less than or equal to a given value, and proposed a heuristic solution based on an iterative process. Touris and Sauer [74] present an approach based on the branch and bound to solve the same problem. Nevertheless, the existing results fail to provide practically effective and computationally efficient methods to analyze and solve the problems in such systems.

The rest of this chapter is structured as follows. Chapter 3.2 presents the problem statement and recall a previous approach proposed in [73]. Chapter 3.3 introduces some liveness conditions for TWMGs. In Chapter 3.4, we propose a heuristic solution for the marking optimization problem under single server semantics based on a live marking. Following the algorithm, an illustrate example is given. Chapter 3.5 proposes a detailed comparison between the proposed approach and a previous one. Conclusions are finally drawn in Chapter 3.6.

3.2 Problem Formulation and Existing Approaches

3.2.1 Problem Formulation

In this chapter, the *marking optimization problem* of a TWMG under single server semantics is considered. The problem consists in finding an initial marking M_0 such that minimizes a weighted function of the initial marking while the cycle time is less than or equal to a given value. We consider a non-negative cost vector $y \in \mathbb{N}^{|P|}$ that is a P-semiflow since the value of $y^T \cdot M$ at every reachable marking $M' \in \mathcal{R}(N, M)$ is an invariant. In particular, if Γ denotes the set of elementary circuits of the net, we can write the cost vector y as the weighted sum of all minimal P-semiflows, i.e.,

$$y = \sum_{\gamma \in \Gamma} \lambda_\gamma \cdot y_\gamma, \tag{3-1}$$

where y_γ denotes the minimal P-semiflow of circuit γ and λ_γ represents the cost of the resources modeled by tokens in the support of y_γ.

The type of resources considered in this thesis are renewable, i.e., the resources are not consumed by the operations and become available again after they have been released, such as machines, tools, and equipments. In addition, we do not necessary consider homogeneous resources. In terms of manufacturing systems, the cost of the resources will remains constant as the production process proceeds.

Problem 3.1. Let $N = (P, T, Pre, Post)$ be a TWMG with a set of the elementary circuits Γ and $y \in \mathbb{N}^{|P|}$ be a non-negative cost vector as defined in Eq. (3-1). Given a positive real number b that represents the upper bound of the cycle time, we look for an initial marking M_0 which minimizes the weighted sum of tokens:

$$\begin{aligned} &\min\ f(M_0) = y^T \cdot M_0 \\ &\text{s.t.} \\ &\ \ \chi(M_0) \le b. \end{aligned} \tag{3-2}$$

□

Proposition 3.1. [73] Under single server semantics, Problem (3-2) has a solution iff

$$b \ge \chi' = \max\{x_i \cdot \delta(t_i), t_i \in T\} \tag{3-3}$$

where x is the minimal T-semiflow and $\delta(t_i)$ is the delay time of transition t_i

3.2.2 A Previous Approach

In this subchapter, we will briefly recall an approach dealing with the marking optimization problem of TWMGs presented by Sauer in [73].

The proposed iterative heuristic algorithm starts with an initial marking M_0 such that

$$M_0(p) = x_{p^\bullet} \cdot Pre(p, p^\bullet),\ \forall p \in P.$$

Obviously, under the condition imposed by Proposition 3.1 this marking is feasible for Eq. (3-2), i.e., it satisfies $\chi(M_0) \le b$.

The approach requires to evaluate the cycle time and the corresponding average marking by simulation. The cycle time is estimated when its value has converged to a preassigned precision.

At each iteration step, one place $p^* \in P$ is selected to remove a token from M_0 as long as the cycle time is less than or equal to the upper bound b. The selected place $p^* \in P$ 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 $M_0(p^*) = 0$, a marking reachable from M_0 containing at least one token in p^* is computed.

Before removing one token from the selected place p, it is necessary to verify that the WMG is going to stay live. If the net is not live after removing one token from $p^* \in P$, they select another place which belongs to $P \backslash \{p^*\}$. 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 huge number of iteration steps to remove the redundant tokens.

3.3 Liveness of TWMGs

Theorem 3.1. [27] A TWMG system $\langle N^{\delta}, M \rangle$ 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 [23, 127]. 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 [94] 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.* [27] and Chrzastowski-Wachtel and Raczunas [24] propose a few methods to verify the liveness of weighted circuits. First they define a weighted function with respect to a marking, i.e.,

$$W(M) = y^T \cdot M, \tag{3-4}$$

where y is a minimal P-semiflow. Furthermore, they define a *greatest dead marking* M_D as:

$$M_D = (v(p_1) - 1, v(p_2) - 1, \ldots, v(p_n) - 1)^T. \tag{3-5}$$

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

Proposition 3.2. [27] If $W(M_0) > W(M_D)$, then the weighted circuit is live.

Less restrictive conditions for liveness also exist. Let $\mathbb{R}^+$ be a set of positive real numbers and $\mathcal{M}(\omega) = \{M|W(M) = \omega, \omega \in \mathbb{R}^+\}$. The *least live weight* is the minimal ω such that $\forall M \in \mathcal{M}(\omega)$, M is a live marking. In [24] the least live weight of a weighted circuit with a minimal P-semiflow y was defined as

$$W_L = W(M_D) - g(y_1, y_2, \cdots, y_n), \tag{3-6}$$

where g is the Frobenius number.[1] 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. (3-6) can be computed and the following proposition holds.

Proposition 3.3. [24] If $g(y_1, y_2, \cdots, y_n)$ has no non-negative integer solution and the marking M_0 satisfies $W(M_0) = W_L$, 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. (3-6).

Example 3.1. Consider a weighted circuit γ in Fig. 3.1, we have $y = (3, 4, 3)^T$, $M_D = (3, 2, 2)^T$, $x = (4, 3, 3)^T$, and $W_L = W(M_D) - g(y_1, y_2, y_3)$ = 23$-g$(3, 4, 3) =23$-$5=18. We can conclude that any marking M with weight $W(M) >$ 23 or $W(M) =$ 18 is a live marking. ◇

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.

3.4 Marking Optimization Under Single Server Semantics

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 cycle time of the TWMG. If the cycle time is greater than the upper bound of the cycle time, we add tokens to some circuits until the cycle time is less than or equal to the upper bound of the cycle time. We select the places

[1]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.

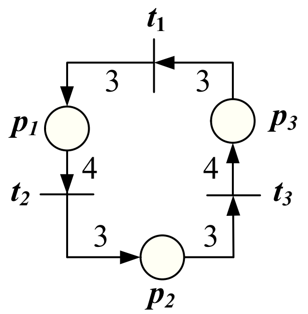

Fig. 3.1 A weighted circuit γ.

to which tokens should be added so as to increase the performance index $f(M_0)$ as small as possible.

3.4.1 Useful tokens

The initial marking $M_0(p)$ of any place p can be replaced by $M_0^\star(p)$ tokens without any influence on the precedence constraints induced by p (see [25] and [26]), where

$$M_0^\star(p) = \left\lfloor \frac{M_0(p)}{\text{gcd}_p} \right\rfloor \cdot \text{gcd}_p \tag{3-7}$$

As a result, we can deduce that the cycle time at M_0 and $M_0^\star$ are the same. However, the value of $f(M_0^\star)$ is less than or equal to $f(M_0)$.

Example 3.2. Consider a TWMG system $\langle N^\delta, M_0\rangle$ shown in Fig. (3.2). The initial marking of the TWMG is $M_0 = (11, 1)^T$ and gcd_{p_1} = gcd_{p_2} = 2.

$$M_0^\star(p_1) = \left\lfloor \frac{M_0(p_1)}{\text{gcd}_{p_1}} \right\rfloor \cdot \text{gcd}_{p_1} = \left\lfloor \frac{11}{2} \right\rfloor \cdot 2 = 10$$
$$M_0^\star(p_2) = \left\lfloor \frac{M_0(p_2)}{\text{gcd}_{p_2}} \right\rfloor \cdot \text{gcd}_{p_2} = \left\lfloor \frac{1}{2} \right\rfloor \cdot 1 = 0$$

Then $M_0^\star = (10, 0)^T$ and we can check that $f(M_0^\star) = 10 < f(M_0) = 12$ and the cycle time at M_0 and $M_0^\star$ are identical, i.e., $\chi(M_0)$ = $\chi(M_0^\star) = 17$. ◇

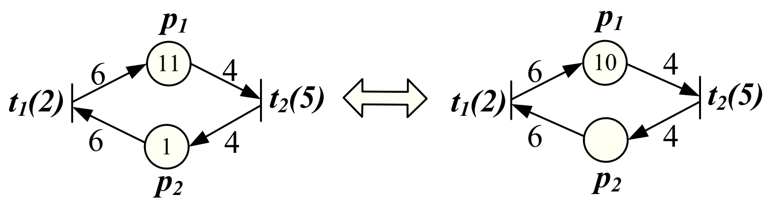

Fig. 3.2 Useful tokens for a TWMG system $\langle N^{\delta}, M\rangle$.

3.4.2 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:

$$\begin{cases} min\ f(M) = y^T \cdot M \\ s.t.\ \ C(M,\gamma)\ \ \forall \gamma \in \Gamma \end{cases} \tag{3-8}$$

where

$$C(M,\gamma) : y_{\gamma}^T \cdot M = W_L^{\gamma} \tag{3-9}$$

or

$$C(M,\gamma) : y_{\gamma}^T \cdot M > W(M_D^{\gamma}) \tag{3-10}$$

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. (3-9) 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. (3-10).

We point out that it may happen that IPP (3-8) has no feasible solution due to the presence of the equality constraints given by Eq. (3-9) that may not be compatible. Should this situation occur, we use for all circuits the inequality constraints given by Eq. (3-10), thus ensuring that a feasible solution exists.

When there exists more than one optimal solution for the marking M, we choose one. Then the initial marking M_0 can be computed using Eq. (3-7), i.e., $M_0 = M^{\star}$. 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 $f(M_0)$ is small. If the cycle time of M_0 is greater than the upper bound of the cycle time, we add tokens to the net until the requirement on the cycle time is satisfied. Otherwise, the initial marking M_0 is a heuristically good solution (although possibly not optimal).

3.4.3 Selection of the Places to Add Tokens

After we select an initial marking M_0, we can compute the cycle time $\chi(M_0)$ of the TWMG and $\chi_\gamma(M_0)$ for every elementary circuit. If the cycle time satisfies the condition $\chi(M_0) \leq b$, no more tokens should be added and the marking M_0 is chosen as a solution.

If the cycle time does not satisfy the condition $\chi(M_0) \leq b$, two situations are possible. If there exist circuits γ's that have cycle time greater than b, i.e., $\chi_\gamma(M_0) > b$, tokens should be added to all these circuits. The set of selected circuits is denoted as $\Gamma_c = \{\gamma \in \Gamma | \chi_\gamma(M_0) > b\}$. However, it may also happen that for any circuit $\gamma \in \Gamma$, $\chi_\gamma(M_0) \leq b$ holds, even if the cycle time of the net is $\chi(M_0) > b$. 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 $\Gamma_c = \{\gamma \in \Gamma | \chi_\gamma(M_0) = \chi^\star(M_0)\}$, where $\chi^\star(M_0)$ 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 $f(M_0)$ as small as possible, i.e., the increment of the criterion value $f(M_0)$ should be the least after adding gcd_{p_r} tokens. We define an n-dimensional vector I of zeros and ones.

$$I^T = (I_{p_1}, I_{p_2}, \cdots, I_{p_n}) \tag{3-11}$$

where

$$I_{p_r} = \begin{cases} 1, & add \ \text{gcd}_{p_r} \ tokens \ to \ place \ p_r \\ 0, & add \ 0 \ token \ to \ place \ p_r \end{cases} \tag{3-12}$$

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

$$P_a = \{p_r | I_{p_r} = 1\} \tag{3-13}$$

and

$$g_d = (\text{gcd}_{p_1} \cdot y_1, \text{gcd}_{p_2} \cdot y_2, \cdots, \text{gcd}_{p_n} \cdot y_n)^T,$$

where y is a P-semiflow of the net and $\text{gcd}_{p_r} \cdot y_r$ represents the increment of $f(M_0)$ after adding gcd_{p_r} tokens to place p_r. We denote by $\Delta f(M_0)$ the total increment of $f(M_0)$, where

$$\Delta f(M_0) = I^T \cdot g_d \tag{3-14}$$

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

$$\begin{cases} \min \ \ \Delta f(M_0) \\ s.t. \ \sum_{p \in \gamma} I_p = 1, \ \forall \gamma \in \Gamma_c \end{cases} \tag{3-15}$$

The constrains in Eq. (6-24) will ensure that only one place should be selected for each circuit.

3.4.4 Heuristic Solution

We can summarize the proposed procedure in Algorithm 1. In step 6 of Algorithm 1, the cycle time needs to be computed. In this chapter, we use the Petri net tool HYPENS [90] to compute the cycle time via simulation.

Algorithm 1: Marking optimization under single server semantics

---

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

Output: An initial marking M_0 such that the cycle time of the net satisfies $\chi(M_0) \leq b$.

1: Compute the marking M_D.
2: For every elementary circuit $\gamma \in \Gamma$, compute $W(M_D^\gamma) = y_\gamma^T \cdot M_D^\gamma$.
3: For every elementary circuit $\gamma \in \Gamma$, compute $W_L^\gamma = y_\gamma^T \cdot M_D^\gamma - g$ if possible.
4: Compute a marking M that satisfies Eq. (3-8).
5: Compute an initial marking $M_0 = M^\star$.
6: Compute the cycle time $\chi(M_0)$ and $\chi_\gamma(M_0)$, $\forall\gamma \in \Gamma$.
7: If $\chi(M_0) \leq b$, stop and $\boldsymbol{M_0}$ is a solution.
8: **While** $\chi(M_0) > b$
{
If $\exists\gamma$, $\chi_\gamma(M_0) > b$,
tokens should be added to all these circuits in $\Gamma_c = \{\gamma \in \Gamma | \chi_\gamma(M_0) > b\}$
Else
$\Gamma_c = \{\gamma \in \Gamma | \chi_\gamma(M_0) = \chi^\star(M_0)\}$;
Compute I and P_a;
Add tokens to P_a and update M_0;
}
9: Output an initial marking M_0.

3.4.5 Case Study

Example 3.3. We consider the TWMG model N^δ in Fig. 3.3. There are four weighted circuits in the TWMG:

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

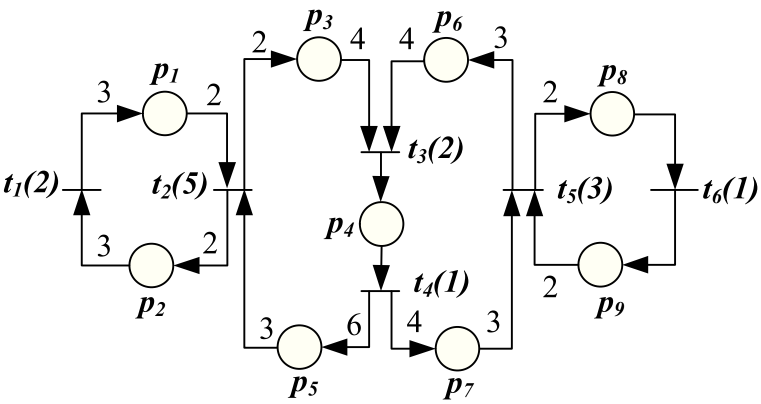

Fig. 3.3 The TWMG model N^{δ} for Example 3.3.

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

$$\begin{cases} x_1 = (2,3,0,0,0,0)^T \\ x_2 = (0,2,1,1,0,0)^T \\ x_3 = (0,0,3,3,4,0)^T \\ x_4 = (0,0,0,0,1,2)^T \end{cases}$$

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

$$\begin{cases} y_1 = (1,1,0,0,0,0,0,0,0)^T \\ y_2 = (0,0,3,12,2,0,0,0,0)^T \\ y_3 = (0,0,0,4,0,1,1,0,0)^T \\ y_4 = (0,0,0,0,0,0,0,1,1)^T \end{cases}$$

The cost of γ_1 and γ_3 is twice the cost of γ_2 and γ_4, i.e., $\lambda_{\gamma_1} = \lambda_{\gamma_3} = 2$ and $\lambda_{\gamma_2} = \lambda_{\gamma_4} = 1$. Therefore, the P-semiflow used in the criterion $f(M_0)$ is $y = 2y_1 + y_2 + 2y_3 + y_4$= (2, 2, 3, 20, 2, 2, 2, 1, 1)^T, and the minimal T-semiflow of the net is x=(4, 6, 3, 3, 4, 8)^T.

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

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, gcd_{p_9} = 1, and

$$g_d = (2, 2, 6, 20, 6, 2, 2, 1, 1)^T$$

min $f(M) = 2M(p_1) + 2M(p_2) + 3M(p_3) + 20M(p_4) + 2M(p_5) + 2M(p_6) + 2M(p_7) +$

$M(p_8) + M(p_9)$

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

We obtain a marking $M = (4, 0, 4, 0, 0, 0, 6, 2, 0)^T$ and the initial marking $M_0 = M^\star = M$. From Table 3.1, we can find that the cycle time of γ_1 and γ_2 are greater than the upper bound of the cycle time b at the initial marking M_0. Then, we compute I^T and P_a to add tokens.

$$\min\ \Delta f(M_0) = 2I_{p_1} + 2I_{p_2} + 6I_{p_3} + 20I_{p_4} + 6I_{p_5} + 2I_{p_6} + 2I_{p_7} + I_{p_8} + I_{p_9}$$

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

Table 3.1 The iteration process for Example 3.3.

M_0	(4, 0, 4, 0, 0, 0, 6, 2, 0)^T	(5, 0, 4, 0, 3, 0, 6, 2, 0)^T	(6, 0, 4, 0, 3, 0, 6, 2, 0)^T
$\chi_{\gamma_1}(M_0)$	38	34	30
$\chi_{\gamma_2}(M_0)$	39	30	30
$\chi_{\gamma_3}(M_0)$	21	21	21
$\chi_{\gamma_4}(M_0)$	20	20	20
$\chi(M_0)$	43	34	30
b	30	30	30
$f(M_0)$	34	42	44
Γ_c	$\{\gamma_1, \gamma_2\}$	$\{\gamma_1\}$	
P_a	$\{p_1, p_5\}$	$\{p_1\}$	

We can find that 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 3.1 that after the first iteration step, $\chi_{\gamma_1}(M_0) > b$ holds. Then, we only need to add tokens to γ_1 to decrease the cycle time. The optimal marking is M = (6, 0, 4, 0, 3, 0, 6, 2, 0)^T and the weight sum of tokens is $f(M)$ = 44. ◇

Example 3.4. Consider the TWMG model N^δ in Fig. 3.4. The marking obtained by Eq. (3-8) is $M = (3, 3, 0, 1, 1)^T$. We have

$$M^\star(p_1) = \left\lfloor \frac{M(p_1)}{\text{gcd}_{p_1}} \right\rfloor \cdot \text{gcd}_{p_1} = \left\lfloor \frac{3}{2} \right\rfloor \cdot 2 = 2$$

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

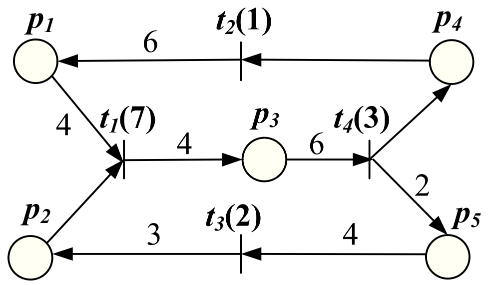

Fig. 3.4 The TWMG model N^{δ} for Example 3.4.

Then the initial marking is $M_0 = M^{\star} = (2, 3, 0, 1, 0)^T$. The iteration process is shown in Table 3.2 and the optimal marking is $(2, 3, 4, 1, 0)^T$. $\diamond$

By enumerating all the possible markings for Examples 3.3 and 3.4, we found that the solutions obtained by our proposed approach are optimal.

Table 3.2 The iteration process for Example 3.4.

M_0	(2, 3, 0, 1, 0)^T	(2, 3, 2, 1, 0)^T	(2, 3, 4, 1, 0)^T
$\chi_{\gamma_1}(M_0)$	29	25	21
$\chi_{\gamma_2}(M_0)$	26	26	21
$\chi(M_0)$	30	26	21
b	21	21	21
$f(M_0)$	20	24	28
Γ_c	$\{\gamma_1, \gamma_2\}$	$\{\gamma_1, \gamma_2\}$	
P_a	$\{p_3\}$	$\{p_3\}$	

3.5 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 [73]. We review this iterative heuristic approach in Chapter 3.2.2 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 subchapter, 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 simulation stops when the system enters a cycle, i.e., we obtain a new state which already exits before, and the cycle time of the system is computed as in Eq. (2-2).

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 $b = max\{x_i \cdot \delta(t_i), t_i \in T\}$. 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 [90] 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 Tables 3.3 and 3.4 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 "o.o.t" in Table 3.3 means that the computation cannot be finished within a reasonable time. As shown in Table 3.4, we can see that the proposed method is much faster than that by Sauer [73] 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 $y^T \cdot M_0$ is greater). The main reason that the proposed approach produces a worse result is that the initial marking computed by Eqs. (3-8) and (3-7) 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.

Table 3.3 Simulation results for the approach of Sauer and the approach proposed in this chapter (He).

				Sauer [ave]			He [ave]		
Nb. of cycles	Nb. of nets	$\lvert P\rvert$ [ave]	$\lvert T\rvert$ [ave]	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
2	10	9	8	64.7	615	34.3	1.9	44	38.5
4	10	15	12	279.7	3676	80.2	3.6	155	85.8
6	10	22	17	387.5	8890	100.8	4	358	114.1
10	10	40	31	o.o.t	o.o.t	o.o.t	4.3	753	191.5

Table 3.4 A comparison between the approach of Sauer and the approach proposed in this chapter (He).

He/Sauer [ave]		
Iteration steps	CPU times [s]	Objective function
26.4%	29.2%	101.0%
9.4%	26.8%	111.3%
2.2%	9.4%	106.7%
1.5%	6.0%	114.1%
4.3/o.o.t	753/o.o.t	191.5/o.o.t

Looking for a better and fast solution, we combine the approach proposed in this chapter with that of Sauer [73], namely He+Sauer, as seen in Table 3.5. First, a candidate marking M_0 is computed by the proposed approach. Then we use the approach of Sauer to remove tokens if possible. The simulation results in Tables 3.5 and 3.6 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 chapter. 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.

Table 3.5 Simulation results for the combined approach (He+Sauer).

				He+Sauer [ave]		
Nb. of cycles	Nb. of nets	$\|P\|$ [ave]	$\|T\|$ [ave]	Iteration steps	CPU time [s]	Obj. fun.
1	10	4	4	4.7	23	29.5
2	10	9	8	4.1	63	34.3
4	10	15	12	6.5	193	80.2
6	10	22	17	8.5	472	100.8
10	10	40	31	11.1	973	167.3

Table 3.6 A comparison between the approach of Sauer and the combined approach (He+Sauer).

He+Sauer/Sauer [ave]			He+Sauer/He [ave]		
Iteration steps	CPU time [s]	Obj. fun.	Iteration steps	CPU time [s]	Obj. fun.
38.9%	40.9%	100.0%	157.9%	155.6%	99.1%
18.6%	35.6%	100.0%	215.0%	140.7%	91.4%
3.9%	11.0%	100.0%	189.1%	138.9%	94.1%
2.9%	7.6%	100.0%	293.7%	137.2%	87.9%
11.1/o.o.t	973/o.o.t	167.3/o.o.t	282.0%	131.0%	87.0%

As one can see, the proposed method needs to find all the elementary circuits and corresponding 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. 3.5, 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 $y = \vec{1}_{2n}$ in the criterion $f(M_0)$. Table 3.7 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.

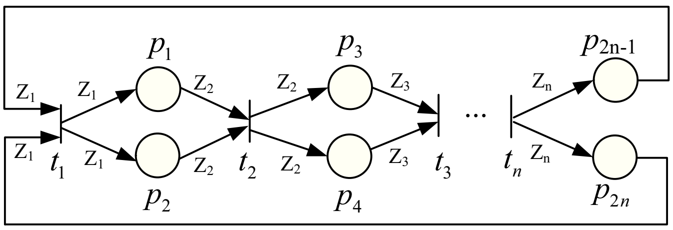

Fig. 3.5 A TWMG model N^δ with a large number of circuits.

Table 3.7 Simulation results for Example 3.5.

	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	o.o.t/128	o.o.t/7320	o.o.t/50
8	o.o.t/140	o.o.t/12194	o.o.t/60
9	o.o.t/o.o.t	o.o.t/o.o.t	o.o.t/o.o.t

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. 3.5 consists in assigning the same number of tokens to the places p_{2i} as in place p_{2i-1}.

3.6 Conclusion

This chapter addresses the problem of marking optimization of a TWMG under single server semantics. The problem consists in finding an initial marking to minimize the weighted sum of tokens in places while the cycle time is less than or equal to a given value.

We propose an iterative heuristic algorithm to solve the marking optimization problem. At each step, we select places from some circuits to which useful tokens are added until the cycle time is less than or equal to the desired value. Numerical simulation studies show that the proposed method requires less iteration steps and thus is much more efficient than the approach in [73]. 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 [73] with a significant reduction of computational costs.

The results presented in this chapter have also been published in:

Z. He, Z. W. Li, and A. Giua, "Optimization of deterministic timed weighted marked graphs," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 1084-1095, 2017.

Chapter 4 Marking Optimization of TWMGs Under Infinite Server Semantics

In this chapter we study the marking optimization problem of deterministic TWMG (Problem (3-2) in Chapter 3) under infinite server semantics, which is a more general case. We show some important properties about the stationary behavior of TWMGs under infinite server semantics. Based on these properties, we develop two heuristic approaches to obtain a near optimal solution. Several examples are presented to illustrate the approach.

4.1 Motivation

Performance optimization of manufacturing systems using time Petri nets have been extensively studied in the literature. The optimization problem is solved by heuristic algorithms such as genetic algorithms, simulated annealing and threshold accepting, and analytical approaches based on solving ILPP.

Wang and Zeng [128] study a time PN model of workflows constrained by resources which are conflict-free nets. They propose a method to verify the risks and found the best implementation case by assuming that all required resources have been prepared well before the start of the activity. However, the proposed method suffers scalability problems when handling large scale systems. By transforming a TWMG into an *equivalent* TMG, Nakamura and Silva [95] develop an algorithm to compute the cycle time of TWMGs under infinite server semantics. The marking optimization problem of TMGs under infinite server semantics is studied by the same author and a tabu search approach is proposed to obtain a near optimal solution [118]. In [96], the behavior of a TWMG is studied by using (min,+) algebra and a linearisation method is proposed to determine the initial marking of a TWMG under restrictive conditions.

As we discussed in Chapter 2, services in a transition are provided sequentially under single server semantics. While under infinite server semantics the number of concurrent servers is equal to the enabling degree of the transition. In fact, single server semantics can be simulated by infinite server semantics adding to each transition a self-loop place with one tokens. For this reason we adopt this more general semantics in this chapter. However, the approach proposed for single server semantics in Chapter 3 fails to tackle this problem under infinite server semantics. From theoretical point of view, there does not exist a lower bound of the cycle time under infinite server semantics, i.e., the cycle time can be as small as possible if we put enough tokes in the system. As a result, Proposition 3.1 dose not hold for TWMGs under infinite server semantics. In addition, the number of iteration steps of Algorithm 1 will be large when the desired value of cycle time is small. Thus, we propose some new heuristic algorithms to solve Problem (3-2) under infinite server semantics.

This chapter is organized in five subchapters. Chapter 4.2 discusses the stationary behavior of TWMGs under infinite server semantics and presents some important properties. In Chapter 4.3, two heuristic approaches are developed to solve the marking optimization problem of TWMGs under single server semantics. Chapter 4.4 presents an illustrative example to show the effectiveness of the proposed approaches. Finally, conclusions are reached in Chapter 4.5.

4.2 Stationary Behavior of TWMGs Under Infinite Server Semantics

This subchapter is devoted to illustrate how the initial distribution of tokens affects the stationary behavior of a TWMG under infinite server semantics. We will show that some important results that hold in the case of TMGs may not hold for this class of nets.

Property 4.1. The cycle time of a cyclic TWMG system $\langle N^{\delta}, M_0 \rangle$ is greater than or equal to the maximal cycle time among all circuits, i.e.,

$$\chi(M_0) \geq \max_{\gamma \in \Gamma} \chi_{\gamma}(M_0). \tag{4-1}$$

It is obvious that the cycle time of a cyclic manufacturing system can not be smaller than the slowest cycle time among all circuits and at most is equal to the slowest one. The following example shows that the cycle time of the system can be greater than that of the slowest circuit.

Example 4.1. Let us consider a cyclic painting process. Machine MA_1 takes one unit of raw material and produces six semi-finished products PR_1 which needs to be painted. Machine MA_2 takes four liters of raw pigment and produces three bags of paint PR_2 (the volume of each is 4/3 liters). Then, Machine MA_3 takes one bag of paint PR_2 and four items of semi-finished product PR_1 and executes the painting process. Finally, a batch transportation device removes six painted product from the workshop and brings one unit of raw material to machine MA_1 and two liters of raw pigment to machine MA_2, respectively.

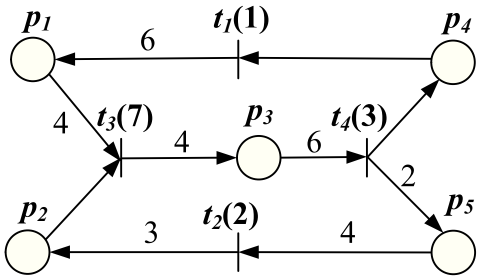

Fig. 4.1 A cyclic painting process.

Table 4.1 Physical meaning of each transition for Example 4.1.

Transition	t_1	t_2	t_3	t_4
physical meaning	MA_1	MA_2	MA_3	transport
Execution times	1	2	7	3

Table 4.2 Physical meaning of each place for Example 4.1.

Places	Physical meaning
p_1	semi-finished product PR_1
p_2	paint PR_2
p_3	product PR_1
p_4	raw material of product PR_1
p_5	raw pigment

This automated cyclic painting process is modelled by a net with four timed transitions: each transition corresponds to a different operation. The TWMG model is shown in Fig. 4.1 and Tables 4.1 and 4.2 present the physical meanings of transitions and places.

There are two elementary circuits $\gamma_1 = p_4t_1p_1t_3p_3t_4$ and $\gamma_1 = p_5t_2p_2t_3p_3t_4$, corresponding to the manufacturing process of PR_1 and PR_2, respectively. The minimal P-semiflow of γ_1 is $y_1 = (1, 0, 1, 6, 0)^T$ and while the minimal P-semiflow of γ_2 is $y_2 = (0, 4, 1, 0, 3)^T$. Thus, we consider a weight vector $y = y_1 + y_2 = (1, 4, 2, 6, 3)^T$. The physical meaning of the weighted vector y is that six items of semi-finished product in p_1 are produced from one item of raw material in p_4 and three packaged paints in p_1 are produced from four items of raw dyestuff in p_5, while four items of painted products in p_3 are manufactured by using four items from p_1 and one item from p_2. Thus, the resources used ratio for each place is equal to y.

Assuming the initial marking of the TWMG is $M_0 = (2, 1, 22, 0, 0)^T$, the cycle time of the system is shown in Table 4.3. The cost of resources used for γ_1 and γ_2 are 24 and 26 and the cycle time of the two circuits are 5.5 and 6. Nevertheless, we find that the cycle time of the system is equal to 7.7 which is greater than the minimal one among the two circuits, i.e.,

Table 4.3 Cycle time analysis for Example 4.2.

Marking	χ_{γ_1}	χ_{γ_2}	χ	$y_1^T \cdot M$	$y_2^T \cdot M$	$y^T \cdot M$
$M_0 = (2, 1, 22, 0, 0)^T$	5.5	6	7.7	24	26	50
$M_1 = (0, 0, 24, 0, 0)^T$	5.5	6	6	24	24	48

$7.7 > \max\{5.5, 6\}$. ◇

Property 4.2. The cycle time of two TWMG systems $\langle N^\delta, M_0\rangle$ and $\langle N^\delta, M_1\rangle$ with same net structure can be different even all the cycle time of their circuits are identical, i.e.,

$$\begin{gathered}\chi(M_0) \neq \chi(M_1), \\ \chi_\gamma(M_0) = \chi_\gamma(M_1),\ \forall\gamma \in \Gamma.\end{gathered} \tag{4-2}$$

We prove this property by showing the following example.

Example 4.2. Consider the net in Fig. 4.1 and assume that the initial marking is $M_1 = (0, 0, 24, 0, 0)^T$. Table 4.3 shows the cycle time analysis of marking M_1. One may find that for marking M_1 the cycle time of each circuit is identical with marking M_0 while the cycle time of the system is greater than that of M_0. ◇

From Example 4.2, we find that the cost of resources used for M_1 is 48 which is smaller than that of M_0, while the cycle time of system is greater than that of M_0. This has practical significance for the cycle time optimization problem which consists in finding an initial marking to maximize the cycle time of the system with a bounded resources. It means that marking M_1 is better than M_0 because it has a smaller resources used. In the following, we will further discuss this problem.

We study the TWMG systems $\langle N^\delta, M_0\rangle$ and $\langle N^\delta, M_1\rangle$ by analyzing the two circuits. Figs. 4.2 and Fig. 4.3 show the marking distribution of γ_1 and γ_1 associate to M_0 and M_1. We have $M_0^{\gamma_1} = (0, 0, 24, 0, 0)$, $M_0^{\gamma_2} = (0, 0, 24, 0, 0)$, $M_1^{\gamma_1} = (2, 0, 22, 0, 0)$, and $M_1^{\gamma_2} = (1, 0, 22, 0, 0)$.

For $M_0^{\gamma_1}$, we can firing transitions $t_4t_1t_3t_3$ in order and obtain a new marking $(0, 0, 24, 0, 0)^T$ which is identical to marking $M_1^{\gamma_1}$, namely, $M_1^{\gamma_1} \in \mathcal{R}(N, M_0^{\gamma_0})$. For $M_0^{\gamma_2}$, transition t_3 can be fired which results in a new marking $(0, 0, 26, 0, 0)^T$ and this new marking has more tokens in p_3 than marking $M_1^{\gamma_1}$. Thus, it seems that the cycle time of marking M_0 should not be smaller than that of marking M_1, which is contrary to the result shown in Table 4.3. The fact is that for the TWMG system t_3 is not enabled because $M_0(p_1) < Pre(p_1, t_1)$. Thus, two tokens in p_1 and one token in p_2 will be trapped, i.e., cannot be used for the system at marking M_0, while no tokens are trapped at marking $M_1^{\gamma_1}$. This is mainly due to the

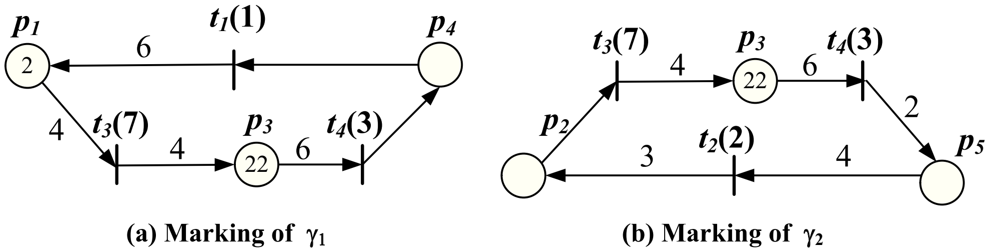

Fig. 4.2 The marking of each circuit under M_0 for Example 4.2.

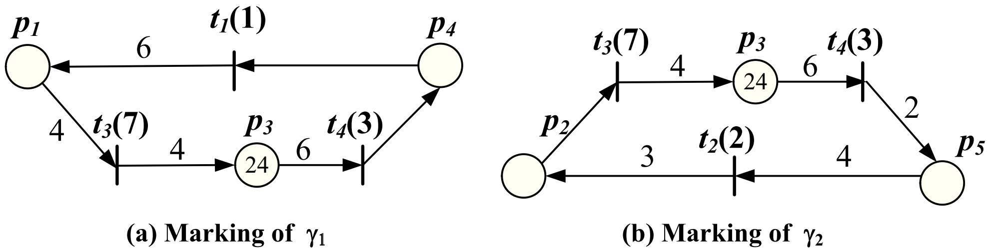

Fig. 4.3 The marking of each circuit under M_1 for Example 4.2.

synchronization of the two circuits. Each of them becomes mutually constrained and results in a lower cycle time of the system.

4.3 Marking Optimization Under Infinite Server Semantics

We propose here two different heuristic solutions to solve the marking optimization problem of a TWMG under infinite server semantics. A candidate live initial marking is firstly computed by an analytical method. The cycle time of this selected marking is usually greater than the desired value b. Thus, more tokens should be added to decrease the cycle time until it satisfies the constraint $\chi(M) \leq b$. In the following, an MILLP method to compute a candidate live initial marking is proposed. Finally, two different heuristic approaches are presented to solve the optimization problem.

4.3.1 Selection of a Candidate Marking

Useful tokens: First, we recall some notations on useful tokens. For a TWMG, the initial marking $M(p_i)$ of any place p_i can be replaced by $M^\star(p_i) = \left\lfloor \frac{M(p_i)}{\text{gcd}_{p_i}} \right\rfloor \cdot \text{gcd}_{p_i}$ tokens without any influence on the precedence constraints induced by p_i.

If $M(p_i)$ is not a multiple of gcd_{p_i}, there will always be $M(p_i) - M^\star(p_i)$ tokens remaining in place p_i that will never be used in the firing of the output transition of place p_i.

As a result, we can deduce that the cycle time at M and $M^{\star}$ are the same.

Selection of a live initial marking: It was shown in [120] and [69] that a lower bound for the cycle time of a live and bounded TWMG system $\langle N^{\delta}, M\rangle$ can be computed by solving following LPP:

$$\begin{array}{l} \min v \\ \text{s.t.} \\ \quad C \cdot z + v \cdot M \geq Pre \cdot \theta \end{array} \tag{4-3}$$

where $\theta \in \mathbb{N}^m$ is the vector containing all firing delays of timed transitions (recall that $m = |T|$). Note that for a TMG whose minimal T-semiflow is equal to $\vec{1}$, thus the element θ_i of vector θ is simple equal to the delay time of corresponding transition t_i, i.e.,

$$\theta = (\delta(t_1), \delta(t_2), \ldots, \delta(t_m))^T.$$

Nevertheless, the vector θ of a TWMG should be modified as follows:

$$\theta = (x_1 \cdot \delta(t_1), x_2 \cdot \delta(t_2), \ldots, x_m \cdot \delta(t_m))^T,$$

where x is the minimal T-semiflow.

The decision variables are $v \in \mathbb{R}^+$ and $z \in \mathbb{R}^m$: the optimal value of v is a lower bound of the cycle time of the TWMG system $\langle N^{\delta}, M\rangle$, i.e.,

$$\chi(M) \geq v. \tag{4-4}$$

To start our heuristic solution, we present an analytical method to select a live initial marking M based on Eq. (6-29). The cycle time of this marking usually satisfy $\chi(M) \leq b$. Nevertheless, we find that in practical examples, the cycle time $\chi(M)$ is very close to the desired value b. Let us first recall some basic results regarding liveness of a WMG. Note that Propositions 1 and 2 are valid for both infinite server semantics and single server semantics.

Proposition 4.1. Let M be a marking which satisfies the following condition:

$$\begin{array}{l} \min v \\ \text{s.t.} \\ \begin{cases} C \cdot z + v \cdot M \geq Pre \cdot \theta, & (a) \\ v = b, & (b) \\ y_{\gamma}^T \cdot M > W(M_D^{\gamma}), \forall \gamma \in \Gamma, & (c) \\ \frac{M(p_i)}{\text{gcd}(p_i)} \in \mathbb{N}, \ \forall p_i \in P. & (d) \end{cases} \end{array} \tag{4-5}$$

Then, the TWMG system $\langle N^{\delta}, M\rangle$ will be live and b is a lower bound of the cycle time $\chi(M)$.

Proof: Constraint (a) is adopted from Eq. (6-29) and can provide a marking M whose lower bound of the cycle time is equal to b if C , Pre, θ, and b are given. The constraint (b) specifies that the lower bound of the cycle time should equal to b, i.e., $\chi(\boldsymbol{M}) \geq b$.

As we discussed in Proposition 3.2, constraint (c) ensures that the TWMG system $\langle N^{\delta}, M\rangle$ will be live. The number of tokens in each place p_i should be a multiple of gcd_{p_i} which is guaranteed by constraint (d). □

As it is stated in Proposition 4.1, the cycle time of M is usually greater than or equal to the upper bound b. Thus, tokens should be added to the net until the requirement on the cycle time is satisfied.

4.3.2 Heuristic Approach 1

The main idea underlying this heuristic approach is the following: at each iteration step, we add tokens to some circuits until the cycle time is less than or equal to the upper bound of the cycle time.

After we obtain a candidate initial marking M_0, we can compute the cycle time $\chi(M_0)$ of the TWMG and $\chi_{\gamma}(M_0)$ for every elementary circuit. Tokens should be added to the circuits which satisfy the following condition:

$$\Gamma_c = \{\gamma \in \Gamma | \chi_{\gamma} > b\}. \tag{4-6}$$

And for each circuit γ belongs to Γ_c, we select one place p_r and add gcd_{p_r} tokens to it. We choose the one that increases $f(M_0)$ as little as possible, i.e., the increment of the criteria value $f(M_0)$ should be the least after adding gcd_{p_r} tokens.

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

$$\begin{aligned} &\min \;\; \Delta f(M_0) \\ &\text{s.t.} \\ &\quad \textstyle\sum_{p\in\gamma} I_p = 1, \; \forall \gamma \in \Gamma_c, \end{aligned} \tag{4-7}$$

where I is an n-dimensional vector as defined in Eq. (3-11) and $\Delta f(M_0)$ represents the total increment of $f(M_0)$ in Eq. (3-14). The set of selected places P_a is defined in Eq. (3-13). The constrains in Eq. (4-7) ensures that only one place should be selected for each circuit that belongs to Γ_c.

4.3.3 Heuristic Approach 2

We propose here another heuristic approach to solve the optimization problem. The basic idea of the heuristic process is to allocate tokens, which reduces the cycle time $\chi(M)$

as much as possible while increases the objective function $f(M)$ (i.e., weighted sum of tokens) as less as possible. At each step, we choose one circuit which has the maximal cycle time (also called critical circuit) among all circuits and add tokens to this circuit. Thus, the selected circuit in Eq. (4-6) should be redefined as follows.

$$\Gamma_c = \{\gamma \in \Gamma | \chi_\gamma(M) = \chi^\star(M)\}, \tag{4-8}$$

where

$$\chi^\star(M) = \max_{\gamma \in \Gamma} \chi_\gamma(M). \tag{4-9}$$

If there exists more than one critical circuit, we choose one. After we choose a critical circuit, we select one place p and add k tokens to it. The number k is a multiple of gcd_p which represents the minimal number of tokens that we should add to decrease the cycle time of the critical circuit. It can be computed by using simulation. We denote the decrease in the cycle time by $\Delta_\gamma \chi(M)$ after allocating k tokens to place p. We have

$$\Delta\chi_\gamma(M) = \chi_\gamma(M') - \chi_\gamma(M), \tag{4-10}$$

where M' is the marking such that $M'(p) = M(p) + k$ and $M'(p') = M(p')$ if $p' \neq p$. Let $\Delta f(M)$ be the gain in criterion value, i.e., the resources that we add, where

$$\Delta f(M) = y_p \cdot k.$$

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

$$\Delta_p = \frac{\Delta f(M)}{\Delta\chi_\gamma(M)}. \tag{4-11}$$

Tokens will be allocated to the place such that

$$P_a = \{p^* | \Delta_{p^*} = \min_{p \in \Gamma_c} \Delta_p\}. \tag{4-12}$$

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 all the optimal allocations to next iteration step.

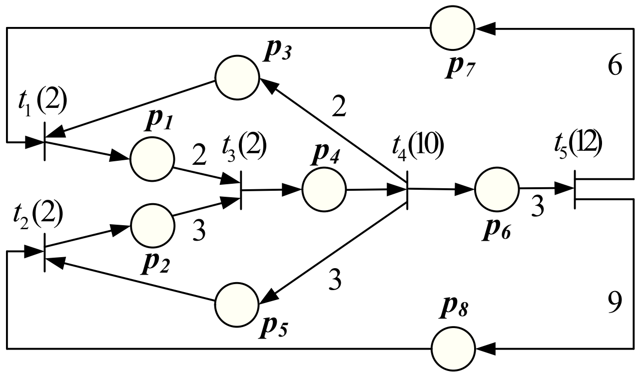

Fig. 4.4 An assembly line.

4.4 Case Study

Example 4.3. Let us illustrate the proposed approaches through an example taken from the literature. It combines cyclic assembly process, buffers, WIP, and batch operations. Two parallel machines (machine one and machine two) are working on items. Machine three loads two parts produced by machine one and three parts produced by machine two and assembles them to get one product. The assembly process is finished by machine four. The batching transportation device removes three finished products from the workshop and brings six items to machine one and nine items to machine two, respectively. The TWMG model of the assembly process is depicted by Fig. 4.4. Transitions t_1, t_2, t_3, t_4, and t_5 represent machine one , machine two, machine three, machine four, and transportation device, respectively.

The minimal T-semiflow of the TWMG is $x_1 = (6, 9, 3, 3, 1)$ and the minimal P-semiflows are $y_1 = (1, 0, 0, 2, 0, 2, 1, 0)^T$, $y_2 = (0, 1, 0, 3, 0, 3, 0, 1)^T$, $y_3 = (1, 0, 1, 2, 0, 0, 0, 0)^T$, and $y_4 = (0, 1, 0, 3, 1, 0, 0, 0)^T$. Thus, the weighted elementary circuits corresponding to all minimal P-semiflows are $\gamma_1 = p_1t_3p_4t_4p_6t_5p_7t_1$, $\gamma_2 = p_2t_3p_4t_4p_6t_5p_8t_2$, $\gamma_3 = p_1t_3p_4t_4p_3t_1$, and $\gamma_4 = p_2t_3p_4t_4p_5t_2$.

The number of tokens in γ_1 and γ_2 represent the number of items proceed (i.e., WIP) by machines one and two, respectively. Thus, we initialize the cost of the tokens in these circuits to two, and that of circuits three and four to one, i.e., $\lambda_1 = 2$, $\lambda_2 = 2$, $\lambda_3 = 1$, and $\lambda_4 = 1$. Thus the P-semiflow we used in the criteria is

$$y = \sum_{\gamma \in \Gamma} \lambda_\gamma \cdot y_\gamma = (3, 3, 1, 15, 1, 10, 2, 2)^T.$$

Table 4.4 Heuristic process of approach 1 for Example 4.3.

step	M	b	$\chi(M)$	$\chi_{\gamma_1}(M)$	$\chi_{\gamma_2}(M)$	$\chi_{\gamma_3}(M)$	$\chi_{\gamma_4}(M)$
0	$(0,0,0,10,0,0,0,0)^T$	8	8.7	8.7	8.7	4.2	4.2
1	$(0,0,0,10,0,0,1,1)^T$	8	8.7	8.7	8.7	4.2	4.2
2	$(0,0,0,10,0,0,2,2)^T$	8	8.7	8.7	8.7	4.2	4.2
3	$(0,0,0,10,0,0,3,3)^T$	8	8.7	8.7	8.7	4.2	4.2
4	$(0,0,0,10,0,0,4,4)^T$	8	8.7	6.5	8.7	4.2	4.2
5	$(0,0,0,10,0,0,4,5)^T$	8	8.7	6.5	8.7	4.2	4.2
6	$(0,0,0,10,0,0,4,6)^T$	8	6.5	6.5	6.5	4.2	4.2

Table 4.5 Heuristic process of approach 2 for Example 4.3.

Step	M	b	$\chi(M)$	$\chi_{\gamma_1}(M)$	$\chi_{\gamma_2}(M)$	$\chi_{\gamma_3}(M)$	$\chi_{\gamma_4}(M)$
0	$(0,0,0,10,0,0,0,0)^T$	8	8.7	8.7	8.7	4.2	4.2
1	$(0,0,0,10,0,0,4,0)^T$	8	8.7	6.5	8.7	4.2	4.2
2	$(0,0,0,10,0,0,4,6)^T$	8	6.5	6.5	6.5	4.2	4.2

Let us consider the following optimization problem

$$\begin{array}{l} \min\ y^T \cdot M \\ \text{s.t.} \\ \quad \chi(M) \leq 8 \end{array}$$

By using the technique introduced in Eq. (4-5), we can obtain an initial marking $M_0 = (0,0,0,10,0,0,0,0)^T$. This marking has a cycle time which is very close to the desired value. Thus, the optimization problem can be efficiently solved by the proposed heuristic approaches. We use HYPENS [90] to implement the heuristic algorithms 1 and 2 and the simulation results are shown in Tables 4.4, 4.6 and 4.5, 4.7 respectively. As one can see, it takes six iteration steps for approach 1 and two iteration steps for approach 2. Note that at each iteration step, heuristic approach 2 needs compute more information than heuristic 1. Both the solutions of approaches 1 and 2 have the same value of objective function $f(M)$, i.e., the total cost of the resources of the assembly line is 170. $\diamond$

To better verify the effectiveness of the two heuristic approaches, we test the example for different value of b and the simulation results are shown in Table 4.8. We can observe that in all the test cases, heuristic approach 1 is slightly faster than heuristic approach 2, while the obtained objective functions $f(M)$ are the same. Note that in the case that $b = 2$, the marking M obtained by the MILPP (4-5) is a heuristically good solution, i.e., $\chi(M) \leq b$. Thus, we do not need to add more tokens to the system. The simulation studies show that the two approaches produce comparable results. They always find the same optimal solution and the execution time is very similar. While the presented results which approach 1 are

Table 4.6 Selection places of heuristic approach 1 for Example 4.3.

step	Γ_c	P_a	$f(M)$	CPU time [s]
0	$\{\gamma_1, \gamma_2\}$	$\{p_7, p_8\}$	150	–
1	$\{\gamma_1, \gamma_2\}$	$\{p_7, p_8\}$	154	–
2	$\{\gamma_1, \gamma_2\}$	$\{p_7, p_8\}$	158	–
3	$\{\gamma_1, \gamma_2\}$	$\{p_7, p_8\}$	162	–
4	$\{\gamma_2\}$	$\{p_8\}$	166	–
5	$\{\gamma_2\}$	$\{p_8\}$	168	–
6	–	–	170	92.7

Table 4.7 Selection places of heuristic approach 2 for Example 4.3.

Step	Γ_c	P_a	Number of tokens k	$f(M)$	CPU time [s]
0	γ_1	p_7	4	150	–
1	γ_2	p_8	6	158	–
2	–	–	–	170	156.5

always faster than approach 2 may depend on the particular example considered.

4.5 Conclusion

In this chapter the marking optimization of a TWMG under infinite server semantics is studied. We provide some properties for cycle time analysis of TWMGs under infinite server semantics. Based on these properties, we develop two heuristic approaches to obtain a near optimal solution. These proposed algorithmes can provide a near optimal solution step by step and also apply for the marking optimization of deterministic TWMGs under single server semantics by adding to each transition a self-loop place with one token.

The results presented in this chapter have also been published in:

Z. He, Z. W. Li, I. Demongodin, A. Giua. "Marking optimization of deterministic timed weighted marked graphs under infinite server semantics", In *Proceedings of the 3rd*

Table 4.8 Simulation results for Example 4.3 with different value of b.

	Heuristic approach 1			Heuristic approach 2		
b	Iteration steps	Objective function $f(M)$	CPU time [s]	Iteration steps	Objective function $f(M)$	CPU time [s]
2	0	585	12.1	0	585	12.1
5	6	261	101.4	2	261	127.6
12	6	125	94.5	2	125	107.4
18	3	86	42.4	2	86	81.3

International Conference on Control, Decision and Information Technologies, (CoDIT'16), 2016: 1-6.

Z. He, Z. W. Li, A. Giua. "Stationary behavior of manufacturing systems modeled by timed weighted marked graphs", In *Proceedings of the IEEE Region 10 Conference* (TENCON'16), 2016: 3374-3377.

Chapter 5 Cycle time Optimization of TWMGs Under Single Server Semantics

In this chapter, the cycle time optimization of deterministic TWMG under single server semantics is studied, which is a dual problem of marking optimization problem. The problem consists in finding an initial marking to minimize the weighted sum of tokens in places while the cycle time is less than or equal to a given value. In addition, we consider single server semantics. To the best of our knowledge, this problem has not been addressed in the literature. We transform a TWMG into several equivalent TMGs and formulate a mixed integer linear programming model to solve this problem. Moreover, several techniques are proposed to reduce the complexity of the proposed method. We show that the proposed method can always find an optimal solution.

5.1 Motivation

Optimization problems are common in the setting of concurrent systems, where a finite set of shared resources must be properly assigned so as to optimize the system performance, i.e., maximize throughput of a manufacturing system, maximize the number of final adopters of a new product. However, these systems are usually large and complex such that it poses difficult problems to find an optimal resource allocation policy to reach their maximal throughput.

Manufacturing systems such as flexible manufacturing systems and automated manufacturing systems can be naturally modeled by Petri nets. The resource optimization problem based on Petri nets has been extensively studied in the literature. For instance, Hee *et al.* [112] and Li and Reveliotis [111] present some methods to compute optimal resource allocation in stochastic PNs. Chen *et al.* [70] develop a new PN model called resource assignment PN to compute the time needed to execute each project under the described scenarios. Rodriguez *et al.* [93] propose a heuristic method to solve the resources optimization problem for process systems with shared resources under the assumption that the considered PNs are live.

By contrast to the aforementioned works, we are interested in resource optimization for TWMGs which are conflict free nets, i.e., there exist no shared resources. This class of Petri nets has usually been used for modeling and analyzing manufacturing systems. Moreover, the TWMGs are not initially assumed to be live, i.e., we need to find a live resources assignment policy which maximizes the throughput of the system.

This chapter is structured as follows. In Chapter 5.2, we present the problem statement. Chapter 5.3 gives an algorithm to transform a TWMG to an equivalent TMG under single server semantics. In Chapter 5.4, we propose an analytical method to solve the optimization problem. Several measures are taken to improve the algorithm to reduce the computational cost in Chapter 5.5. Moreover, we study a more general optimization problem. Some experimental results are presented in Chapter 5.6. Conclusions are finally drawn in Chapter 5.7.

5.2 Problem Formulation

In this chapter, the *cycle time optimization* (also called the maximum throughput initial state assignment problem) of a TWMG under single server semantics is studied. We aim to find an initial marking M such that the weighted sum of tokens in places is less than or equal to a given value. Among all feasible solutions, we look for those that minimize the cycle time, i.e., maximize the throughput.

We consider a non-negative cost vector $y \in \mathbb{N}^{|P|}$ as defined in Eq. (3-1) that is a P-semiflow, i.e.,

$$y = \sum_{\gamma \in \Gamma} \lambda_\gamma \cdot y_\gamma,$$

where y_γ denotes the minimal P-semiflow of circuit γ and λ_γ represents the cost of the resources modeled by tokens in the support of y_γ. The value of $y^T \cdot M$ at every reachable marking $M' \in \mathcal{R}(N, M_0)$ is an invariant.

Problem 5.1. Let N^δ be a TWMG with a set of the elementary circuits Γ and $y \in \mathbb{N}^{|P|}$ be a non-negative cost vector as defined in Eq. (3-1). Given a positive real number R that represents the *upper bound* on the cost of resources, we look for an initial marking M_0 which minimizes the cycle time $\chi(M_0)$:

$$\begin{aligned} &\min\ \chi(M_0) \\ &\text{s.t.} \\ &\quad y^T \cdot M_0 \leq R. \end{aligned} \tag{5-1}$$

□

5.3 Transformation from a TWMG to an equivalent TMG Under Single Server Semantics

One way to analytically compute the cycle time of a TWMG is to convert it into an equivalent TMG. In fact, Munier [94] shows that a TWMG system $\langle N^\delta, M\rangle$ can be transformed into an *equivalent* TMG system $\langle \hat{N}^\delta, \hat{M}\rangle$ which describes the same precedence constraints on the firing of transitions. This implies that the cycle time[1] of the two systems is identical, i.e.,

$$\chi(M) = \chi(\hat{M}).$$

This equivalent TMG system depends on the initial marking M and the minimal T-semiflow x of the TWMG. Since it is necessary for us to use this transformation method, we present it in Algorithm 1. All notations in the algorithm are from previous definitions and $x_{out(p_i)}$ in Eq. (5-3) (resp., $x_{in(p_i)}$ in Eq. (5-5)) represents the elementary T-semiflow component corresponding to transition $t_{out(p_i)}$ (resp., $t_{in(p_i)}$). Note that Eqs. (5-3) and (5-5) admit only one solution (a_s, b_s and c_s, d_s) for each value of s.

Example 5.1. Consider a TWMG model N^δ in Fig. 5.1. We assume that the initial marking is $M_0 = (0, 0, 4)^T$.

Transformation of transitions: The minimal T-semiflow is $x = (1, 2, 1)^T$. Then the transitions t_1, t_2 and t_3 are replaced by one transition, two transitions and one transition, respectively. Moreover, places q's to connect these transitions are added. The nets drawn by dotted lines in Fig. 5.2 correspond to the intra transition sequential systems.

Transformation of places: Since place p_1 satisfies the condition $w(p_1) > v(p_1)$, it is replaced by place p_1^1 according to Algorithm 2. We compute a_1 and b_1 to determine the marking and structure of place p_1^1. Places p_2 and p_3 satisfy the condition $w(p_i) \leq v(p_i)$ ($i = 2, 3$), and then places p_2 and p_3 are replaced by p_2^1 and p_3^1, respectively. Markings and structures of p_2^1 and p_3^1 are computed by Eq. (5-5).

Fig. 5.2 shows the equivalent TMG with the initial marking $M = (0, 0, 4)^T$. There are totally four transitions and seven places. ◇

The structure of the equivalent TMG (i.e., the arcs connecting places and transitions) depends on the marking M of the TWMG. However, this dependence is periodic as shown in the following proposition.

[1] In the following, we will denote by $\chi(M)$ the cycle time of a TWMG system $\langle N^\delta, M\rangle$ and by $\chi(\hat{M})$ the cycle time of the equivalent TMG system $\langle \hat{N}^\delta, \hat{M}\rangle$.

Algorithm 2: Transformation of a TWMG into a TMG under single server semantics

Input: A TWMG system $\langle N^\delta, M\rangle$.

Output: An equivalent TMG system $\langle \hat{N}^\delta, \hat{M}\rangle$ such that $\chi(M) = \chi(\hat{M})$.

1: Compute the minimal T-semiflow $x = (x_1, \ldots, x_m)^T$ of net N.

2: *(Transformation of transitions).* Replace each transition $t_i \in T$ by x_i transitions, t_i^1, t_i^2, …, $t_i^{x_i}$, with the same firing delay of t_i. These transitions are connected by an elementary circuit with all weights equal to 1. Add x_i places q_i^1, q_i^2, …, $q_i^{x_i}$, where q_i^a, $a = 1, \ldots, x_i - 1$, is a place connecting transition t_i^a to transition t_i^{a+1} and $q_i^{x_i}$ is a place connecting transition $t_i^{x_i}$ to t_i^1. Only place $q_i^{x_i}$ contains one token and the other places are empty, i.e.,

$$\begin{cases} \hat{M}(q_i^a) = 0,\ \forall i = 1, \ldots, m,\ \forall a = 1, \ldots, x_i - 1, \\ \hat{M}(q_i^{x_i}) = 1,\ \forall i = 1, \ldots, m, \end{cases} \tag{5-2}$$

Thus there exist m mono-marked circuits that are called *intra transition sequential systems*. They do not depend on the initial marking.

3: *(Transformation of places: case 1).* Replace each place $p_i \in P$ such that $w(p_i) > v(p_i)$ by $n_i = x_{in(p_i)}$ places p_i^s, where for $s = 1, \ldots, n_i$:

$$\begin{cases} a_s \cdot x_{out(p_i)} + b_s = \left\lfloor \frac{M(p_i) + w(p_i)\cdot(s-1)}{v(p_i)} \right\rfloor + 1, \\ b_s \in \{1, \ldots, x_{out(p_i)}\}, \\ a_s \in \mathbb{N}. \end{cases} \tag{5-3}$$

Place p_i^s connects transition $t_{in(p_i)}^s$ to transition $t_{out(p_i)}^{b_s}$ and contains a_s tokens, i.e.,

$$\hat{M}(p_i^s) = a_s. \tag{5-4}$$

4: *(Transformation of places: case 2).* Replace each place $p_i \in P$ such that $w(p_i) \leq v(p_i)$ by $n_i = x_{out(p_i)}$ places p_i^s, where for $s = 1, \ldots, n_i$:

$$\begin{cases} c_s \cdot x_{in(p_i)} + d_s = \left\lceil \frac{s \cdot v(p_i) - M(p_i)}{w(p_i)} \right\rceil, \\ d_s \in \{1, \ldots, x_{in(p_i)}\}, \\ c_s \in \mathbb{Z}_{\leq 0}. \end{cases} \tag{5-5}$$

Place p_i^s connects transition $t_{in(p_i)}^{d_s}$ to transition $t_{out(p_i)}^s$ and contains $-c_s$ tokens, i.e.,

$$\hat{M}(p_i^s) = -c_s. \tag{5-6}$$

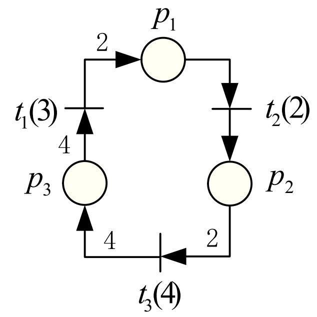

Fig. 5.1 The TWMG model N^δ for Examples 5.1 and 5.3.

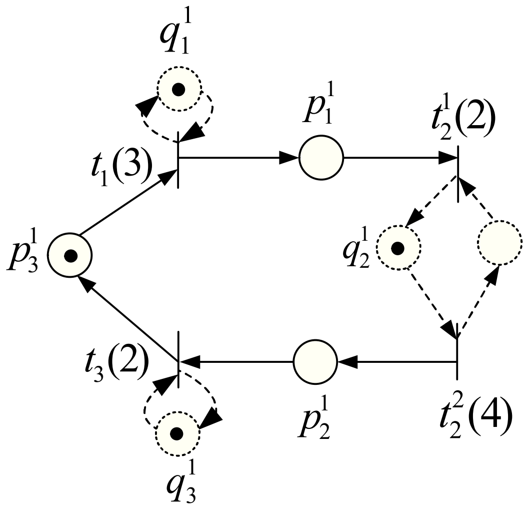

Fig. 5.2 TMG equivalent to the TWMG in Fig. 5.1 for Example 5.1.

Proposition 5.1. Consider a TWMG N^δ with minimal T-semiflow $x = (x_1, x_2, \ldots, x_m)^T$ and two possible initial markings M_1 and M_2. Let $\langle \hat{N}_1^\delta, \hat{M}_1 \rangle$ (resp., $\langle \hat{N}_2^\delta, \hat{M}_2 \rangle$) be the equivalent TMG obtained by Algorithm 1 with input $\langle N^\delta, M_1 \rangle$ (resp., $\langle N^\delta, M_2 \rangle$).

If for a place $p_i \in P$

$$M_2(p_i) = M_1(p_i) + \xi \cdot v(p_i) \cdot x_{out(p_i)} \quad with\ \xi \in \mathbb{N},$$

then the structure corresponding to p_i in $\hat{N}_1^\delta$ and $\hat{N}_2^\delta$ is the same and the markings of the transformed places p_i^s corresponding to p_i in Eqs. (5-4) and (5-6) satisfy

$$\hat{M}_2(p_i^s) = \hat{M}_1(p_i^s) + \xi. \tag{5-7}$$

Proof: Since $M_1(p_i) + x_{in(p_i)} \cdot w(p_i) - x_{out(p_i)} \cdot v(p_i) = M_1(p_i)$, we have $x_{in(p_i)} \cdot w(p_i) = x_{out(p_i)} \cdot v(p_i)$. If $w(p_i) > v(p_i)$, $x_{in(p_i)} < x_{out(p_i)}$ and $s = 1, \ldots, x_{in(p_i)}$, for marking $M_1(p_i)$

of place p_i, it holds that:

$$\begin{cases} a_s \cdot x_{out(p_i)} + b_s = \left\lfloor \frac{M_1(p_i)+w(p_i)\cdot(s-1)}{v(p_i)} \right\rfloor + 1, \\ b_s \in \{1, \ldots, x_{out(p_i)}\},\ a_s \in \mathbb{N}, \end{cases}$$

and for marking $M_2(p_i)$ of place p_i,

$$\begin{cases} a'_s \cdot x_{out(p_i)} + b'_s = \\ \left\lfloor \frac{M_1(p_i)+\xi\cdot x_{out(p_i)}\cdot v(p_i)+w(p_i)\cdot(s-1)}{v(p_i)} \right\rfloor + 1, \\ b'_s \in \{1, \ldots, x_{out(p_i)}\}, \\ a'_s \in \mathbb{N}, \end{cases}$$

$$a'_s \cdot x_{out(p_i)} + b'_s = \left\lfloor \frac{M_1(p_i) + w(p_i)\cdot(s-1)}{v(p_i)} \right\rfloor + \xi \cdot x_{out(p_i)} + 1,$$

then

$$a'_s \cdot x_{out(p_i)} + b'_s = (a_s + \xi) \cdot x_{out(p_i)} + b_s,$$

and

$$\begin{cases} a'_s = a_s + \xi, \\ b'_s = b_s, \end{cases} \tag{5-8}$$

If $w(p_i) \leq v(p_i)$, $x_{in(p_i)} \geq x_{out(p_i)}$ and $k = 1, \ldots, x_{out(p_i)}$, we can obtain the following equation

$$\begin{cases} c'_s = c_s - \xi, \\ d'_s = d_s, \end{cases} \tag{5-9}$$

where a_s and $-c_s$ represent the number of tokens in equivalent places and b_s and d_s represent the structure (input arc or output arc) of equivalent places. According to Eqs. (6-16) and (5-9), it follows that the equivalent structures of $M_1(p_i)$ and $M_2(p_i)$ are identical while

$$\hat{M}_2(p_i^s) = \hat{M}_1(p_i^s) + \xi.$$

□

The previous result implies that the structure corresponding to place p_i in the equivalent TMG is periodic with regard to $M(p_i)$ and the period ϕ_i is equal to $v(p_i) \cdot x_{out(p_i)}$ (or equivalently $w(p_i) \cdot x_{in(p_i)}$).

Example 5.2. Consider a TWMG model N^δ in Fig. 5.3 whose minimal T-semiflow is x=(2, 3)^T. Fig. 5.4 shows the equivalent TMG systems $\langle \hat{N}^\delta, \hat{M} \rangle$ corresponding to different initial markings.

Transitions t_1 and t_2 are replaced by two transitions (t_1^1 and t_1^2) and three transitions (t_2^1, t_2^2, t_2^3), respectively. Places q's to connect these transitions are added. For place p_1 (resp.,

p_2), it is replaced by two (resp., two) places p_1^1 and p_1^2 (resp., p_2^1 and p_2^2). For different initial markings, the structures of equivalent transitions (gray blocks) are always the same, while the structures and markings of equivalent places (blue blocks) may change.

From Proposition 5.1, we can compute the period of each place $\phi_1 = 2$, and $\phi_2 = 6$. The equivalent TMG structures corresponding to M_1, M_3, and M_4 are the same as shown in Figs. 5.4(a), 5.4(c), and 5.4(d). For the marking $M' = (6\xi_1, 6\xi_2)^T$, one can easily check that the structure of the equivalent TMG is identical to that of the net in Fig. 5.4(a) while the markings of equivalent places are $\hat{M}'(p_1^1) = \hat{M}'(p_1^2) = \xi_1$, and $\hat{M}'(p_2^1) = \hat{M}'(p_2^2) = \xi_2$. $\diamond$

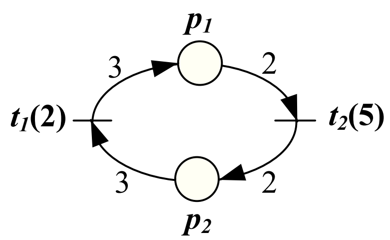

Fig. 5.3 The TWMG net N^{δ} for Example 5.2.

The size of the equivalent TMG is[2] $O(|x|_1)$. More precisely the number of transitions is $\hat{m} = |x|_1$ and that of places is $\hat{n} = \sum_{i=1}^{n} n_i + |x|_1$ which is less than or equal to $2|x|_1$. Theoretically $|x|_1$ can grow exponentially with respect to the net size. However, one finds that in practical examples, this is a quite reasonable number.

5.4 Cycle Time Optimization Under Single Server Semantics

We propose here an MILPP to solve the cycle time optimization problem for TWMGs. We first give some conditions under which the optimization problem admits a finite solution. Then, we show the general idea on which our approach is based. Some techniques are introduced in subchapter 5.4.3 to reduce the equivalent TMGs structures. Finally, we formulate the proposed approach in subchapter 5.4.4.

5.4.1 Existence of Finite Solutions

For a TWMG, the complexity of checking liveness and determining the minimal number of tokens that ensures the liveness remains open. Based on Proposition 3.2 and Theorem 3.1, we present a sufficient condition concerning the existence of a finite solution to the considered optimization problem.

[2]Here $|x|_1$ denotes the 1-norm of T-semiflow x.

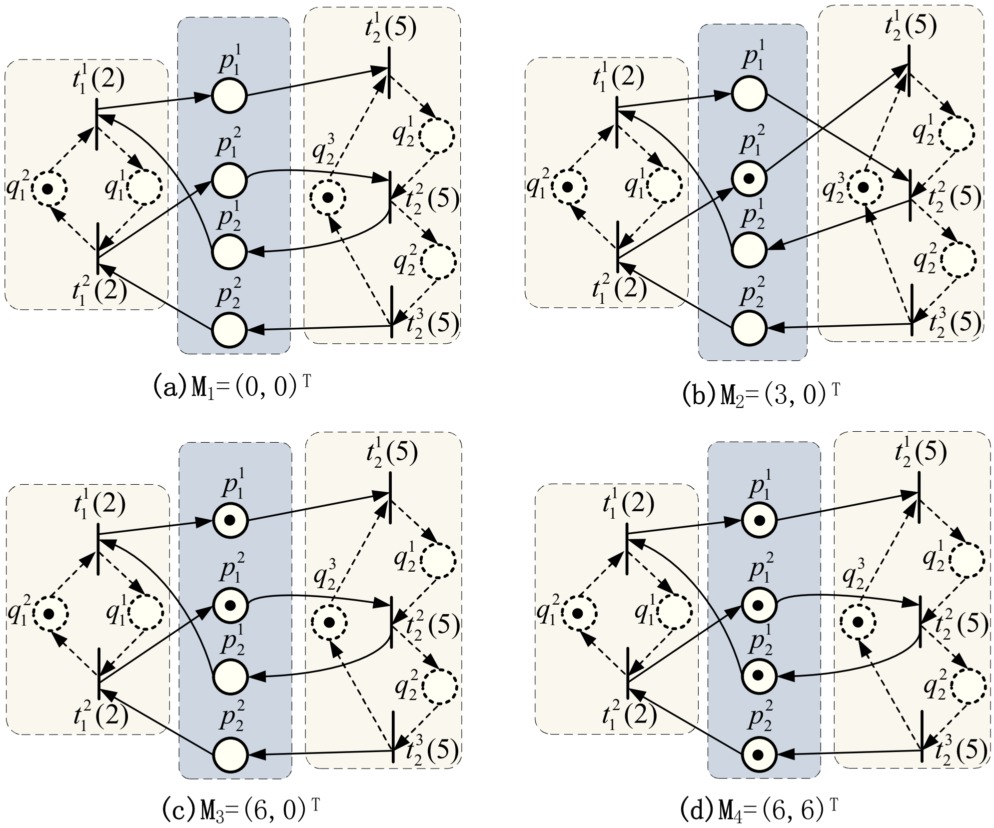

Fig. 5.4 The equivalent TMG systems corresponding to different initial markings for Example 5.2.

Proposition 5.2. Let M be a marking of a TWMG, $y = \sum_{\gamma \in \Gamma} \lambda_\gamma \cdot y_\gamma$ be a cost vector as defined in Eq. (3-1), M_γ^D be the greatest dead marking of circuit γ as defined in Eq. (3-5), and R be a positive real number that represents the upper bound on the cost of resources. Problem (5-1) has a finite solution if $R \geq R^*$, where R^* is a positive real number such that:

$$\begin{aligned} R^* = \quad & \min \quad y^T \cdot M, \\ & \text{s.t.} \\ & \quad y_\gamma^T \cdot M > y_\gamma^T \cdot M_\gamma^D \quad (\forall \gamma \in \Gamma). \end{aligned}$$

Proof: If for any $\gamma \in \Gamma$, $y_\gamma^T \cdot M > y_\gamma^T \cdot M_\gamma^D$ holds, we conclude that each circuit of the TWMG is live according to Proposition 3.2. Then, the TWMG is necessarily live according to Theorem 3.1 and its cycle time will be finite. □

In [73], the author proves that the lower bound of the cycle time under single server semantics is

$$\chi' = \max\{x_i \cdot \delta(t_i), t_i \in T\}, \tag{5-10}$$

where x is the minimal T-semiflow and $\delta(t_i)$ is the delay time of transition t_i.

5.4.2 General Idea

Giua *et al.* [120] present that for a TMG the solution[3] of Problem (5-1) can be computed by solving the following MILPP:

$$\begin{array}{l} \max\ \beta \\ \text{s.t.} \\ \begin{cases} C \cdot \alpha - Pre \cdot \delta \cdot \beta + M \geq 0 \\ y^T \cdot M \leq R \end{cases} \end{array} \tag{5-11}$$

with variables $M \in \mathbb{N}^n$, $\beta \in \mathbb{R}^+$ and $\alpha \in \mathbb{R}^m$. It provides the optimal solution M and the corresponding maximal throughput β (i.e., the inverse of cycle time 1/$\chi(M)$), and α has no obvious physical meaning.

For TWMGs one way to find the optimal solution of Problem (5-1) is to enumerate all possible equivalent TMGs and solve an MILPP (5-11) for each of them to find a marking which has the maximal throughput. However, there are two main problems.

- The number of TMG structures equivalent to a TWMG may be very large. This issue is addressed in Chapter 5.4.3.

- We have to add in Eq. (5-11) a series of constraints to ensure the marking $\hat{M}$ that we find for a given net structure $\hat{N}$ is consistent with the marking M of the original TWMG. We discuss this issue in Chapter 5.4.4.

5.4.3 Reduction of Equivalent TMG structures

According to Proposition 5.1, for each place $p_i \in P$ of a TWMG system $\langle N^\delta, M\rangle$, the structure corresponding to place p_i in the equivalent TMG is periodic with respect to $M(p_i)$ and the period is ϕ_i. Thus, we should compute the equivalent structures for initial marking $M(p_i) = 0, 1, \ldots, \phi_i - 1$.

We note that the set of possible markings of place p_i can be partitioned into ϕ_i subsets such that

$$\bigcup_{k_i=0}^{\phi_i-1} \mathcal{M}_{p_i}^{k_i} = \mathbb{N},\ where\ \{k_i + \xi \cdot \phi_i | \xi \in \mathbb{N}\} \ = \mathcal{M}_{p_i}^{k_i}, \tag{5-12}$$

and all makings of p_i in the same partition $\mathcal{M}_{p_i}^{k_i}$ correspond to the same equivalent structure.

For each place $p_i \in P$, we define $\mathcal{N}_i = \{0, \ldots, \phi_i - 1\}$. Then the set of markings of a

[3] The MILPP in Eq. (5-11) provides a solution under infinite server semantics while here we consider single server. However, the equivalent TMGs constructed by Algorithm 2 are such that the enabling degree of transitions is at most equal to one: this means that their behavior is the same under both infinite and single server semantics.

TWMG can be partitioned into several subsets

$$\bigcup_{(k_1,\ldots,k_n)\in\mathcal{N}_1\times\cdots\times\mathcal{N}_n} \mathcal{M}_{p_1}^{k_1} \times \mathcal{M}_{p_2}^{k_2} \times \ldots \times \mathcal{M}_{p_n}^{k_n} = \mathbb{N}^n. \tag{5-13}$$

For each vector $\boldsymbol{k} = (k_1, \ldots, k_n) \in \mathcal{N}_1 \times \cdots \times \mathcal{N}_n$ corresponding to partition $\mathcal{M}_{p_1}^{k_1} \times \mathcal{M}_{p_2}^{k_2} \times \ldots \times \mathcal{M}_{p_n}^{k_n}$, the equivalent TMGs for all markings in this partition are the same. The total number of such structures (i.e., partitions) is

$$\Phi = \prod_{p_i\in P} \phi_i. \tag{5-14}$$

Note that the number of equivalent structures given by Eq. (5-14) is usually large. We look for more efficient solutions that only require to consider a subset of these structures (i.e., partitions). To reach this goal, the following result is useful.

Lemma 5.1. [25] For a WMG, the initial marking $M(p_i)$ of any place p_i can be replaced by $M^\star(p_i) = \left\lfloor \frac{M(p_i)}{gcd_{p_i}} \right\rfloor \cdot gcd_{p_i}$ tokens without any influence on the precedence constraints induced by p_i.

In fact, if $M(p_i)$ is not a multiple of gcd_{p_i}, there will always be $M(p_i) - M^\star(p_i)$ tokens remaining in place p_i that will never be used in the firing of the output transition of place p_i. As a result, we can deduce that the cycle time at M_0 and $M_0^\star$ are the same.

Example 5.3. Consider the TWMG model N^δ in Fig. 5.1. We assume the initial marking $M_0 = (0, 0, 11)^T$.

$$M_0^\star(p_3) = \left\lfloor \frac{M_0(p_3)}{gcd_{p_3}} \right\rfloor \cdot gcd_{p_3} = \left\lfloor \frac{11}{4} \right\rfloor \cdot 4 = 8$$

Then $M_0^\star = (0, 0, 8)^T$ and we can check that the equivalent TMGs of M_0 and $M_0^\star$ are the same, which implies that the cycle times of system $\langle N, M_0\rangle$ and system $\langle N, M_0^\star\rangle$ are identical, i.e., $\chi(M_0) = \chi(M_0^\star)$. ◇

From Lemma 5.1, when looking for an optimal solution for Problem (5-1), we may restrict our analysis to the markings that belong to a restricted number of partitions where the token content of each place p_i is a multiple of gcd_{p_i}. Hence the number of meaningful subsets in Eq. (5-12) can be reduced as follows:

$$\begin{gathered} \bigcup_{k_i=0}^{\frac{\phi_i}{gcd_{p_i}}-1} \bar{\mathcal{M}}_{p_i}^{k_i} \subseteq \mathbb{N}, \\ \bar{\mathcal{M}}_{p_i}^{k_i} = \{k_i \cdot gcd_{p_i} + \xi \cdot \phi_i | \xi \in \mathbb{N}\}. \end{gathered} \tag{5-15}$$

We define $\bar{\mathcal{N}}_i = \{0, \ldots, \frac{\phi_i}{gcd_{p_i}} - 1\}$ and the set of markings of a TWMG in Eq. (5-13) can be redefined as

$$\mathcal{M}_{opt} = \bigcup_{(k_1,\ldots,k_n)\in\mathcal{N}_1'\times\cdots\times\mathcal{N}_n'} \bar{\mathcal{M}}_{p_1}^{k_1} \times \bar{\mathcal{M}}_{p_2}^{k_2} \times \ldots \times \bar{\mathcal{M}}_{p_n}^{k_n} \subseteq \mathbb{N}^n \tag{5-16}$$

where the number of partitions is reduced to

$$\Phi' = \prod_{p_i\in P} \frac{\phi_i}{gcd_{p_i}}. \tag{5-17}$$

In the following, for the sake of simplicity, we rename the partitions defined in Eq. (5-16) and write

$$\mathcal{M}_{opt} = \bigcup_{j=1}^{\Phi'} \mathcal{M}_j \tag{5-18}$$

where

$$\mathcal{M}_j = \bar{\mathcal{M}}_{p_1}^{k_{j,1}} \times \bar{\mathcal{M}}_{p_2}^{k_{j,2}} \times \ldots \times \bar{\mathcal{M}}_{p_n}^{k_{j,n}} \tag{5-19}$$

i.e., partition j is characterized by the n-tuple $(k_{j,1}, \ldots, k_{j,n})$.

Consider the example in Fig. 5.1. We have $gcd_{p_1} = 1$, $gcd_{p_2} = 1$, $gcd_{p_3} = 4$, $\phi_1 = 2$, $\phi_2 = 2$, and $\phi_3 = 4$. The number of partitions is $\Phi = 16$, while the number of meaningful partitions is $\Phi' = 4$, which is significantly smaller.

5.4.4 Optimal Approaches

We now show how it is possible to solve Problem (5-1) by assuming that the unknown initial marking M of the TWMG belongs to a generic partition $\mathcal{M}_j$ shown in Eq. (5-19).

In this case, due to the special equivalent structure of a marking $M \in \mathcal{M}_j$ in Eq. (5-15), Problem (5-1) can be rewritten as

$$\begin{array}{l} \min\ \chi(M) \\ \text{s.t.} \\ \begin{cases} y^T \cdot M \leq R, \\ M(p_i) = k_{j,i} \cdot gcd_{p_i} + \xi_{j,i} \cdot \phi_i,\ \forall p_i \in P, \\ \xi_{j,i} \in \mathbb{N}, \end{cases} \end{array}$$

We define the vector $\boldsymbol{\xi}_j = (\xi_{j,1}, ..., \xi_{j,n})^T$ and for each place p_i with an initial marking

$$M(p_i) = k_{j,i} \cdot gcd_{p_i},\ k_{j,i} = 0, \ldots, \frac{\phi_i}{gcd_{p_i}} - 1, \tag{5-20}$$

we compute

- the equivalent structure of place p_i, i.e., places $p_i^1, \ldots, p_i^{n_i}$,

- the initial markings correspond to Eq. (5-20), i.e., $\mu_j(p_i^1) = \hat{M}(p_i^1), \ldots, \mu_j(p_i^{n_i}) = \hat{M}(p_i^{n_i})$.

Thus for each partition $\mathcal{M}_j$ given in Eq. (5-19), we can compute the equivalent net structure $\hat{N}_j$, incidence matrix $\hat{C}_j$ and pre-incidence $\hat{Pre}_j$.

Proposition 5.3. For each partition $\mathcal{M}_j$ in Eq. (5-19), we consider the following MILPP

$$\begin{aligned}
&\max\ \beta_j \\
&\text{s.t.} \\
&\begin{cases}
\hat{C}_j \cdot \hat{\alpha}_j - \hat{Pre}_j \cdot \hat{\delta}_j \cdot \beta_j + \hat{M}_j \geq 0, & (a) \\
y^T \cdot M_j \leq R, & (b) \\
M_j(p_i) = k_{j,i} \cdot gcd_{p_i} + \xi_{j,i} \cdot \phi_i,\ \forall p_i \in P, & (c) \\
\hat{M}_j(p_i^s) = \mu_j(p_i^s) + \xi_{j,i}, s = 1, \ldots, n_i, & (d) \\
\hat{M}_j(q_i^a) = 0,\ \forall i = 1, \ldots, m,\ \forall a = 1, \ldots, x_i - 1, & (e) \\
\hat{M}_j(q_i^{x_i}) = 1,\ \forall i = 1, \ldots, m, & (f) \\
\xi_{j,i} \in \mathbb{N}, & (g)
\end{cases}
\end{aligned} \tag{5-21}$$

with variables[4] $\beta_j \in \mathbb{R}_{\geq 0}$, $M_j \in \mathbb{N}^n$, $\hat{M}_j \in \mathbb{N}^{\hat{n}}$, $\hat{\alpha}_j \in \mathbb{R}^{\hat{m}}$, and $\xi_j \in \mathbb{N}^n$. Let $(\beta_j^*, M_j^*, \hat{M}_j^*, \hat{\alpha}_j^*, \xi_j^*)$ be an optimal solution of Eq. (5-21). Thus M_j^* is also an optimal solution of Problem (5-1) restricted to partition $\mathcal{M}_j$.

Proof: The constraint (a) adopted from Eq. (5-11) can provide an optimal solution if $\hat{C}_j$, $\hat{Pre}_j$ and $\hat{\delta}_j$ are given. The constraint (b) specifies that the weighted sum of tokens in places cannot exceed the upper bound on the cost of resources, and the constraint (c) specifies that feasible markings should be restricted to partition $\mathcal{M}_j$.

As shown in Eqs. (6-1) and (6-12), the marking $\hat{M}_j$ of the equivalent TMG should be consistent with the marking M_j of the TWMG; this is ensured by constraints (d), (e) and (f).

In [120] the authors prove that the MILPP can obtain an optimal solution for the cycle time optimization problem. Thus, $(\beta_j^*, M_j^*, \hat{M}_j^*, \hat{\alpha}_j^*, \xi_j^*)$ is an optimal solution of Problem (5-1) restricted to partition $\mathcal{M}_j$. □

Note that the MILPP in Eq. (5-21) has $|x|_1 + n + 1$ variables and at most $6|x|_1 + n + 1$ constrains, where n denotes the number of places of a TWMG.

Property 5.1. Any marking M that produces a cycle time $\chi(M) = \chi'$ as defined in Eq. (5-10) and satisfies $y^T \cdot M \leq R$ is an optimal solution.

[4]Recall that $\hat{n}$ (resp., $\hat{m}$) is the number of places (resp., transitions) of the equivalent TMG.

Proof: According to Eq. (5-10), once we obtain a marking M which has the cycle time $\chi(M) = \chi'$, no more reduction can be obtained no matter how many resources we increase. Obviously, the throughput is maximal and M is an optimal initial marking. □

We can summarize the proposed procedure in Algorithm 3.

Algorithm 3: An MILPP method for the cycle time optimization of a TWMG

Input: A cyclic TWMG N^δ, an upper bound R of its weighted sum of tokens and a P-semiflow y.

Output: An optimal marking M with throughput β such that the weighted sum of tokens satisfies $y^T \cdot M \leq R$.

1. Compute the meaningful partitions of each place in Eq. (5-15).
2. Compute the partitions $\mathcal{M}_{opt}$ of initial marking in Eq. (5-16).
3. $j := 1$, $\beta' := 1/\chi'$, and $\beta := 0$.
4. **while** $j \leq \Phi'$ & $\beta < \beta'$ **do**
    - Transform the TWMG system $\langle N_j, M_j \rangle$ into the equivalent TMG system $\langle \hat{N}_j, \hat{M}_j \rangle$ as shown in Algorithm 2;
    - Compute an optimal marking M_j^* and the corresponding throughput β_j^* for $\langle \hat{N}_j, \hat{M}_j \rangle$ as in Eq. (5-21);
    - **if** $\beta_j^* > \beta$ **then**
        - $\beta := \beta_j^*$;
        - $M := M_j^*$;
    - $j := j + 1$;
5. Output an optimal marking M and the corresponding throughput β.

Proposition 5.4. The output of Algorithm 3 provides an optimal solution for Problem (5-1).

Proof: It is obvious that if we solve Eq. (5-21) for each partition, among all the optimal solutions, we can obtain the maximal throughput

$$\beta = \max_{j=1,\ldots,\Phi'} \beta_j^*,$$

and the corresponding marking M. The global optimal solutions of Problem (5-1) are M and $\chi(M) = 1/\beta$. □

The mechanism of Algorithm 3 can be explained by Fig. 5.5. From a theoretical point of view, we should compute the solutions for all Φ' partitions. However, in practical if we find a marking M whose cycle time converges to the lower bound, there is no need to do more computations. According to Property 5.1, we can conclude that marking M is an optimal solution.

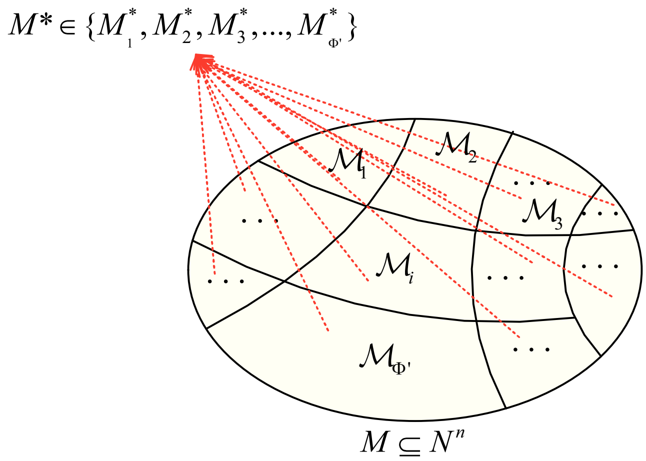

Fig. 5.5 Mechanism of Algorithm 3.

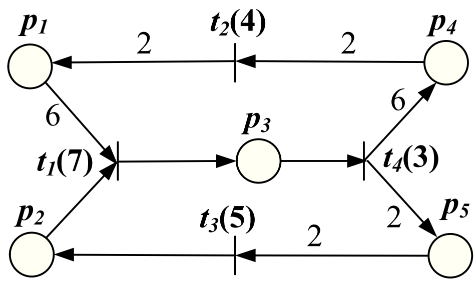

Fig. 5.6 The TWMG model N^δ for Example 5.4.

Example 5.4. Consider the TWMG model N^δ in Fig. 5.6. The minimal T-semiflows is $x = (1, 3, 1, 1)^T$, while the minimal P-semiflows are $y_1 = (1, 0, 6, 1, 0)^T$ and $y_2 = (0, 2, 2, 0, 1)^T$. Therefore, we choose the P-semiflow $y = y_1 + y_2 = (1, 2, 8, 1, 1)^T$. We have $\phi_1 = 6$, $\phi_2 = 1$, $\phi_3 = 1$, $\phi_4 = 6$, $\phi_5 = 2$, $gcd_{p_1} = 2$, $gcd_{p_2} = 1$, $gcd_{p_3} = 1$, $gcd_{p_4} = 2$, and $gcd_{p_5} = 2$. The number of variables is equal to 12 and the number of constrains is equal to 28. The markings of the TWMG are partitioned into $\Phi' = 9$ subsets.

$$\begin{cases}
\mathcal{M}_1 = (6\xi_{1,1}, \xi_{1,2}, \xi_{1,3}, 6\xi_{1,4}, 2\xi_{1,5})^T \\
\mathcal{M}_2 = (2 + 6\xi_{2,1}, \xi_{2,2}, \xi_{2,3}, 6\xi_{2,4}, 2\xi_{2,5})^T \\
\mathcal{M}_3 = (4 + 6\xi_{3,1}, \xi_{3,2}, \xi_{3,3}, 6\xi_{3,4}, 2\xi_{3,5})^T \\
\mathcal{M}_4 = (6\xi_{4,1}, \xi_{4,2}, \xi_{4,3}, 2 + 6\xi_{4,4}, 2\xi_{4,5})^T \\
\mathcal{M}_5 = (6\xi_{5,1}, \xi_{5,2}, \xi_{5,3}, 4 + 6\xi_{5,4}, 2\xi_{5,5})^T \\
\mathcal{M}_6 = (2 + 6\xi_{6,1}, \xi_{6,2}, \xi_{6,3}, 2 + 6\xi_{6,4}, 2\xi_{6,5})^T \\
\mathcal{M}_7 = (2 + 6\xi_{7,1}, \xi_{7,2}, \xi_{7,3}, 4 + 6\xi_{7,4}, 2\xi_{7,5})^T \\
\mathcal{M}_8 = (4 + 6\xi_{8,1}, \xi_{8,2}, \xi_{8,3}, 2 + 6\xi_{8,4}, 2\xi_{8,5})^T \\
\mathcal{M}_9 = (4 + 6\xi_{9,1}, \xi_{9,2}, \xi_{9,3}, 4 + 6\xi_{9,4}, 2\xi_{9,5})^T
\end{cases}$$

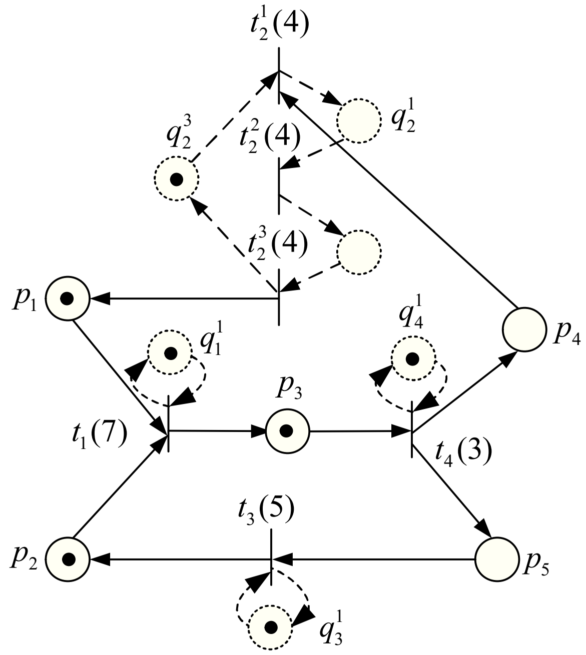

Fig. 5.7 The equivalent TMG system $\langle \hat{N}^{\delta}, \hat{M} \rangle$ for Example 5.4.

Let $R = 20$ be the upper bound on the cost of resources and problems of the form (5-21) can be immediately formulated for each partition $\mathcal{M}_i$ ($i = 1, \ldots, 9$). In the following equation, we will show the MILPP for partition $\mathcal{M}_1$.

$$
\begin{aligned}
&\max \beta_1 \\
&\text{s.t.} \\
&\begin{cases}
\hat{C}_1 \cdot \hat{\alpha}_1 - \hat{Pre}_1 \cdot \hat{\delta}_1 \cdot \beta_1 + \hat{M}_1 \geq 0, & (a) \\
y^T \cdot M_1 \leq 20, & (b) \\
M_1(p_1) = 6\xi_{1,1},\ M_1(p_2) = \xi_{1,2},\ M_1(p_3) = \xi_{1,3}, & \\
M_1(p_4) = 6\xi_{1,4},\ M_1(p_5) = 2\xi_{1,5}, & (c) \\
\hat{M}_1(p_1^1) = \xi_{1,1},\ \hat{M}_1(p_2^1) = \xi_{1,2},\ \hat{M}_1(p_3^1) = \xi_{1,3}, & \\
\hat{M}_1(p_4^1) = \xi_{1,4},\ \hat{M}_1(p_5^1) = \xi_{1,5}, & (d) \\
\hat{M}_1(q_2^1) = 0,\ \hat{M}_1(q_2^2) = 0, & (e) \\
\hat{M}_1(q_1^1) = 1,\ \hat{M}_1(q_2^3) = 1,\ \hat{M}_1(q_3^1) = 1,\ \hat{M}_1(q_4^1) = 1, & (f) \\
\xi_{1,i} \in \mathbb{N},\ i = 1, ..., 5. & (g)
\end{cases}
\end{aligned}
\tag{5-22}
$$

The solutions of Eq. (5-22) are $\beta_1 = 0.083$ and $M_1 = (6, 1, 1, 0, 0)^T$ and the equivalent TMG is depicted in Fig. 5.7. We can observe from Table 5.1 that the cycle time $\chi(M_1)$ is equal to the lower bound χ'. According to Property 5.1, this solution is also globally optimal. $\diamond$

Table 5.1 Optimal solution for Example 5.4.

M_1	β	$\chi(M_1)$	χ'	$y^T \cdot M_1$	R
$(6,1,1,0,0)^T$	0.083	12	12	16	20

5.5 Extension of the Basic Approach

The aim of this subchapter is to further improve the basic approach presented in Chapter 5.4 by reducing the computation complexity of Algorithm 3 and by considering a slightly more general cycle time optimization problem. We first prove that the number of equivalent TMG structures to be analysed can be further reduced by exploring the net structure. Then, we discuss the more general optimization problem.

5.5.1 Further Reduction of Equivalent TMG Structures

In this subsection, we will study the possibility to further reduce the number of equivalent TMG structures in Eq. (5-17).

Example 5.5. Consider the TWMG model N^δ in Fig. 5.3. The periods of places p_1 and p_2 are $\phi_1 = 6$ and $\phi_2 = 6$, respectively. Thus, the number of equivalent TMG structures is $\Phi' = 36$. However, if the number of tokens in place p_1 satisfies the condition $M(p_1) \geq 2$, we can always fire t_2 as many times as possible. Then, we can restrict our attention to partitions satisfying $M(p_1) < 2$, i.e., $M(p_1) = 0$ or $M(p_1) = 1$. As a result, to find the optimal solution, we need only study 12 equivalent TMG structures rather than 36. In the following, some propositions are given to reduce the partition as much as possible. $\diamond$

Proposition 5.5. Let N^δ be a TWMG consisting of only one circuit and described by the following sequence: $p_1 \xrightarrow{v(p_1)} t_1 \xrightarrow{w(p_2)} p_2 \xrightarrow{v(p_2)} \cdots \xrightarrow{v(p_n)} t_n \xrightarrow{w(p_1)} p_1$. The number of partitions can be reduced to

$$\Phi'' = \frac{\phi_n}{gcd_{p_n}} \times \prod_{i=1}^{n-1} \frac{v(p_i)}{gcd_{p_i}}. \tag{5-23}$$

Proof: For any live marking M, we fire at M the transition t_1 as many times as we can. Next we fire t_2 as many times as we can, and so on, until we fire the transition t_{n-1} leaving on the place p_n the maximal number of tokens that can be put without firing t_n.

Then, we obtain a new marking M' such that the number of tokens in each place is $M'(p_i) < v(p_i)$ $(i = 1, \ldots, n-1)$, i.e.,

$$M'(p_i) \in \{0, gcd_{p_i}, \ldots, v(p_i) - gcd_{p_i}\},\ i = 1, \ldots, n-1, \tag{5-24}$$

and $M'(p_n) \in \mathbb{N}$.

As a consequence, the number of equivalent TMG structures can be reduced to Eq. (5-23). By comparing Eq. (5-23) with Eq. (5-17), we can conclude that $\Phi'' \leq \Phi'$ is true due to $v(p_i) \leq \phi_i = v(p_i) \cdot x_{out(p_i)}$. □

Now we will discuss how to reduce the partitions of a TWMG which consists of more than one single circuit. In the following, these notations are used.

- T^*: the set of transitions of N which have only one input place.
- P^*: the set of pre-places of T^*.

Proposition 5.6. Let N^δ be a TWMG with n places and m transitions. The number of partitions of N^δ in Eq. (5-17) can be reduced to

$$\Phi''' = \prod_{p_j \in P \setminus P^*} \frac{\phi_j}{gcd_{p_j}} \times \prod_{p_i \in P^*} \frac{v(p_i)}{gcd_{p_i}}. \tag{5-25}$$

Proof: Let place p_i belong to P^* and transition $t_{out(p_i)}$ be the output transition of p_i. Thus, p_i is the decisive place of transition t, i.e., the firing of $t_{out(p_i)}$ is only decided by p_i. We fire $t_{out(p_i)}$ as many times as possible and the final marking of place p_i will satisfy the condition $M(p_i) < v(p_i)$. The number of meaningful subsets in Eq. (5-15) will be

$$\begin{aligned} &\bigcup_{k_i=0}^{\frac{v(p_i)}{gcd_{p_i}}-1} \bar{\mathcal{M}}_{p_i}^{k_i} \subseteq \mathbb{N},\ \forall p_i \in P^* \\ &\bar{\mathcal{M}}_{p_i}^{k_i} = k_i \cdot gcd_{p_i}. \end{aligned} \tag{5-26}$$

□

As a result, the number of partitions to be considered when searching for an optimal solution is reduced from Φ' in Eq. (5-17) to Φ''' in Eq. (5-25).

5.5.2 A More General Optimization Problem

In many cases, it may be useful to introduce an additional criterion for Problem (5-1) so as to select, among all the solutions that provide the same optimal value of cycle time $\chi(M)$, those that also minimize the total weighted sum of tokens in the net. This problem has practical significance: under a given upper bound on the resources we aim to maximize the throughput and achieve this goal with a minimal cost.

Problem 5.2. Let N^δ be a TWMG and $y \in \mathbb{N}^{|P|}$ be a cost vector as defined in Eq. (3-1). Given a positive real number R that represents the upper bound on the cost of resources and a small positive number $w \in \mathbb{R}^+$, we look for an initial marking M_0 which satisfies the following condition:

$$\begin{aligned} &\min\ \chi(M_0) + w \cdot y^T \cdot M_0 \\ &\text{s.t.} \\ &\quad y^T \cdot M_0 \leq R \end{aligned} \tag{5-27}$$

□

Note that $w \in \mathbb{R}^+$ should be sufficiently small so as to maintain the minimization of $\chi(M)$ as the prior requirement.

By substituting the objective function in Eq. (5-21) with the following function:

$$\max\ \beta_j - w \cdot y^T \cdot M_j, \tag{5-28}$$

the optimal solution of Problem (5-27) can be found. Note that to solve Problem (5-27), we need to compute all the local optimal solutions and compare both the throughput and the cost of resources.

5.6 Experimental Study and Discussion

In this subchapter, two types of experimental results are provided. First, we test the proposed approach on a model of flexible manufacturing system (FMS) taken from the literature [73]. Second, a series of randomly generated nets are explored and the numerical results are given in Chapter 5.6.2. For the application of Algorithm 3, MATLAB has been used with the MILPP toolbox YALMIP [129] on a PC with Pentium Dual-Core CPU 3.0 GHz and 2 GB memory.

5.6.1 Optimization of a Flexible Manufacturing System

The TWMG model N^δ of an FMS is shown in Fig. 5.8. This system is composed of three machines $\mathcal{U}_1$, $\mathcal{U}_2$ and $\mathcal{U}_3$. The manufacturing system is cyclic and 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:

$$\begin{cases} \mathcal{R}_1 : (\mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3) \\ \mathcal{R}_2 : (\mathcal{U}_2, \mathcal{U}_1) \end{cases}$$

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.

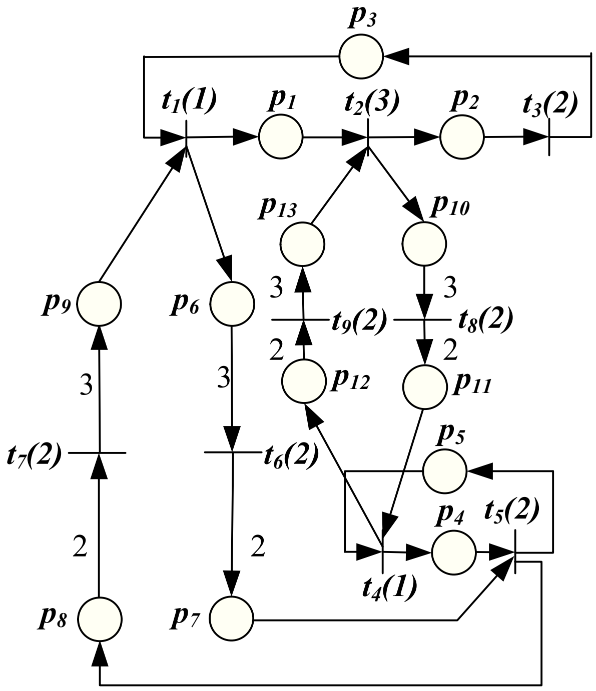

Fig. 5.8 A flexible manufacturing system.

- *Command circuits*: Model the control of the system. One command circuit is associated with each machine $\mathcal{U}_1$ and $\mathcal{U}_2$ to specify that they are cyclically used in both processes.

- *Mixed circuits*: these circuits are partially composed of parts of the command circuits and parts of the process circuits.

The Petri model N^δ in Fig. 5.8 is a strongly connected TWMG with $n = |P| = 13$ and $m = |T| = 9$. There are six elementary circuits :

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

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, for the command circuit γ_3 in Fig. 5.8, we assume that $M(p_{10}) = 0$, $M(p_{11}) = 0$, $M(p_{12}) = 0$, and $M(p_{13}) = 3$ and this command circuit cannot be allocated tokens any more.

The number of tokens in process circuits γ_1 and γ_2 represents that of available pallets for products, i.e., work in process. Thus, the cost of the resources in these circuits are the main economic consumption of the FMS. Tokens belonging to command circuits γ_3 and γ_4 represent information. We have $\lambda_1 = 10$, $\lambda_2 = 10$, $\lambda_3 = 1$, $\lambda_4 = 1$, $\lambda_5 = 1$, and $\lambda_6 = 1$.

The P-semiflow is $y = \sum_{\gamma\in\Gamma} \lambda_\gamma \cdot y_\gamma = (12, 12, 12, 13, 13, 4, 6, 6, 4, 4, 6, 6, 4)^T$ and the minimal T-semiflow is x=(3, 3, 3, 2, 2, 1, 1, 1, 1)^T. We have $\phi_1 = 3$, $\phi_2 = 3$, $\phi_3 = 3$, $\phi_4 = 2$, $\phi_5 = 2$, $\phi_6 = 3$, $\phi_7 = 2$, $\phi_8 = 2$, $\phi_9 = 3$, $\phi_{10} = 3$, $\phi_{11} = 2$, $\phi_{12} = 2$, $\phi_{13} = 3$, $gcd_{p_i} = 1$ ($i = 1, \ldots, 13$) and $P^* = \{p_2, p_6, p_8, p_{10}, p_{12}\}$. Let us consider the following optimization problem:

$$\begin{aligned}
&\min\ \chi(M) + w \cdot y^T \cdot M\\
&\text{s.t.}\\
&\begin{cases} y^T \cdot M \leq 100 \\ M(p_{13}) = 3, \\ M(p_i) = 0,\ i = 10, 11, 12 \end{cases}
\end{aligned}$$

Now, the form represented by Eq. (5-21) can be immediately formulated. The number of variables is equal to 31 and the number of constrains is equal to 90. The markings of the TWMG are partitioned into $\Phi' = 3888$ subsets. According to Proposition 9 proposed in Section 5.5, the number of partitions can be reduced to $\Phi''' = 1296$.

We assume that the upper bound on the cost of resource is $R = 100$ and choose $w = 10^{-8}$. By using Algorithm 3, we find an optimal solution $M = (2, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 3)^T$ and $\chi(M) = 11$ by considering the number of reduced partition Φ'''. It implies that the actual usage of money is 74 and the cost of pallets is 50.

5.6.2 Test of Random Nets

We present some numerical results for Problems (5-1) and (5-27) in Table 5.2 and Table 5.3, respectively. All the tested nets are randomly generated under the assumption that each circuit has at least two places and at most six places and the weight of each arc (resp., delay of each transition) is a random integer number picked up from the interval $[1, 6]$ (resp., $[1, 10]$). For each of them we initialize the upper bound on the cost of resource R to a positive number which is much bigger than R^*. We mention that if R is a number that closes to R^*, the complexity of Problem (5-1) will be the same of Problem (5-27). Using the number of reduced partitions Φ''', we obtain the optimal solutions for Problems (5-1)

Table 5.2 Computation results for Problem (5-1) in terms of different input nets.

					Problem (5-1)		
N	$\|P\|$	$\|T\|$	Φ'	Φ'''	Throughput β	Cost of resources $y^T \cdot M$	CPU time
1	5	4	128	32	0.0476	24	4s
2	8	7	4,608	384	0.0278	104	7s
3	13	9	3,888	1296	0.0909	212	15s
4	17	14	73,728	1536	0.0139	128	20s
5	27	22	10,077,696	20736	0.025	134	30s
6	35	29	5.4e+11	65536	0.0069	196	42s

Table 5.3 Computation results for Problem (5-27) in terms of different input nets.

					Problem (5-27)		
N	$\|P\|$	$\|T\|$	Φ'	Φ'''	Throughput β	Cost of resources $y^T \cdot M$	CPU time
1	5	4	128	32	0.0476	16	41s
2	8	7	4,608	384	0.0278	26	20 min
3	13	9	3,888	1296	0.0909	74	1 h
4	17	14	73,728	1536	0.0139	42	1.9h
5	27	22	10,077,696	20736	0.025	46	27h
6	35	29	5.4e+11	65536	o.o.t.	o.o.t.	o.o.t.

and (5-27). For all the cases in Tables 5.2 and 5.3, we consider the cardinalities of P and T (net size), the number of partition Φ', the number of reduced partition Φ''', the throughput β, the cost of resources $y^T \cdot M$, and the CPU time for Problems (5-1) and (5-27). Note that "o.o.t" (out of time) in Table 5.3 means that the solution of net 6 for Problem (5-27) cannot be found within 48 hours.

As one can see, the solutions for Problem (5-1) can be obtained within a very short time. Once we compute an initial marking which makes the system reach its upper bound of the throughput, the algorithm will stop. Nevertheless, for Problem (5-27) which maximizes the throughput and minimizes the cost of resources, we need to explore all the local optimal solutions and compare both the throughput and the cost of resources. Among all the solutions that provide the same optimal value of throughput, we also want to reduce the cost of resources as much as possible. As a consequence, the computational time will be significantly longer than that of Problem (5-1). When the net size becomes larger, we cannot obtain a solution of Problem (5-27) within a reasonable time. However, the model represented by a TWMG is much smaller than that generated with a TMG. In practical examples, the net size of a TWMG is quite reasonable thanks to the weight.

5.7 Conclusion

This chapter deals with the cycle time optimization problem of deterministic TWMGs under single server semantics. The problem consists in finding an initial marking to minimize the cycle time while the weighted sum of tokens in places is less than or equal to a given value. To the best of our knowledge, this problem has not been addressed in the literature. We transform a TWMG into several equivalent TMGs and formulate a mixed integer linear programming problem solution from the study in [120] to compute an optimal initial marking. The conversion of the obtained marking for the equivalent TMG to a marking associated with the TWMG is presented. Some techniques are introduced to reduce the computational burden of computing the solution. It is shown that, in some cases, we do not need to enumerate all the possible structures to find the optimal solution.

More general allocation problems are studied in the second part of this chapter: among all the solutions that provide the same optimal value of throughput, we aim to obtain the one that also minimizes the cost of resources. The proposed method can also guarantee the convergence to the optimum.

The results presented in this chapter have also been published in:

Z. He, Z. W. Li, and A. Giua, "Cycle time optimization of deterministic timed weighted marked graphs by transformation," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 4, pp. 1318-1330, 2017.

Chapter 6 Cycle time Optimization of TWMGs Under Infinite Server Semantics

In this chapter, we discuss the cycle time optimization problem of deterministic TWMG (Problem (5-1) in Chapter 5) under infinite server semantics, which is a more general case than the one we discussed in Chapter 5. We consider the transformation of a given TWMG into an equivalent *place timed marked graph* (PTMG) proposed by Nakamura and Silva [95] and prove that this transformation is periodical with regard to the initial marking. This allow us to transform a TWMG into a finite family of equivalent PTMGs, each one valid for a partition of set of initial markings. Then, we present an MILPP to solve the optimization problem that requires finding an optimal allocation for the equivalent PTMG under the constraint that the initial marking belongs to a particular partition. In addition, two sub-optimal approaches are proposed in order to reduce the computational complexity.

6.1 Motivation

The cycle time optimization problem of TMGs under infinite server semantics is considered in [120]. Three different approaches which take the full advantage of MILPP are developed to find an optimal schedule. Nakamura and Silva [95] study the cycle time computation of a TWMG under infinite server semantics. While in Chapter 5, we consider the cycle time optimization of TWMGs under single server semantics, in this chapter we study the same problem under infinite server semantics, i.e., the degree of self-concurrency of each transition is infinite. From a physical point of view, the server semantics can be interpreted as the number of times that an operation can be executed concurrently. Under single server semantics, the same operation can only be executed once at a time, while the same operation can be executed as many times as the number of available servers under infinite server semantics.

Inspired by the works in [95], we show that the TWMG can be transform into a finite family of equivalent PTMGs and present an MILPP to solve the optimization problem that requires finding an optimal allocation for the equivalent PTMG under the constraint that the initial marking belongs to a particular partition. Nevertheless, the computational complexity of this approach is very high. Then, a sub-optimal approach called place subset allocation (PSA) is proposed which assigns tokens to a subset of places instead of taking all the places into consideration. Finally, we develop another sub-optimal approach called throughput upper bound (TUB) which does not need to transformation a TWMG into a finite family of equivalent PTMGs. In some practical instances, reduction of the computational cost is very important and necessary. Thus, the presented sub-optimal approaches try to cope with this requirement.

This chapter is organized as follows. Chapter 6.2 recalls a method that transform a TWMG into an equivalent PTMG under infinite server semantics [95]. Chapter 6.3 introduces an MILPP to solve Problem (5-1) under infinite server semantics. Chapter 6.4 presents two sub-optimal solutions to reduce the computational cost. In Chapter 6.5, applications of the proposed approaches and numerical studies are investigated. Conclusions are finally reached in Chapter 6.6.

6.2 Transformation From a TWMG to an Equivalent PTMG Under Infinite Server Semantics

Nakamura and Silva [95] proved that a *transition timed* WMG system $\langle N^\delta, M\rangle$ under infinite server semantics can be transformed into an equivalent *place timed marked graph* (PTMG) system $\langle \hat{N}^\delta, \hat{M}\rangle$. The Petri net language of the PTMG system is the same as that of the TWMG system. This means that the cycle time of the two systems are identical, i.e., $\chi(M) = \chi(\hat{M})$.

Note that the equivalent PTMG system $\langle \hat{N}^\delta, \hat{M}\rangle$ depends on the initial marking M and the minimal T-semiflow x of the TWMG. All notations in Algorithm 4 are defined in previous definitions and $x_{out(p_i)}$ (resp., $x_{in(p_i)}$) denotes the elementary T-semiflow component corresponding to $t_{out(p_i)}$ (resp., $t_{in(p_i)}$).

The transformation of transitions does not depend on the initial marking of the TWMG. The structure of the equivalent PTMG (i.e., the input and output arcs of equivalent places) depends on the initial marking M of the TWMG. Let n_i be the number of equivalent places corresponding to place p_i. The number of equivalent transitions is $\hat{m} = |x|_1$ and that of places is $\hat{n} = \sum_{i=1}^{n} n_i + |x|_1$.

Example 6.1. Consider the TWMG system $\langle N^\delta, M\rangle$ in Fig. 6.1 whose initial marking is $M_0 = (4, 2, 0)^T$. The minimal T-semiflow of the TWMG is $x = (2, 2, 3)^T$ and its corresponding equivalent PTMG system $\langle \hat{N}^\delta, \hat{M}\rangle$ is shown in Fig. 6.2 according to Algorithm 4.

First, each transition t_i ($i = 1, 2, 3$) is replaced by a circuit contains x_i transitions

Algorithm 4: Transformation of a TWMG into a PTMG under infinite server semantics

Input: A TWMG system $\langle N^\delta, M\rangle$

Output: An equivalent PTMG system $\langle \hat{N}^\delta, \hat{M}\rangle$ such that $\chi(M) = \chi(\hat{M})$.

Calculate the minimal T-semiflow $x = (x_1, \ldots, x_m)^T$ of net N^δ.
for each transition $t_i \in T$ **begin**
Replace each transition t_i by x_i transitions, t_i^1, t_i^2, …, $t_i^{x_i}$.
Place q_i^a is added which connects transition t_i^a to transition $t_i^{a \bmod x_i + 1}$ $(a = 1, \ldots, x_i)$.

$$\begin{cases} \hat{M}(q_i^a) := 1,\ \delta(q_i^a) := 0.\ (a = x_i) \\ \hat{M}(q_i^a) := 0,\ \delta(q_i^a) := 0.\ (\text{otherwise}) \end{cases} \tag{6-1}$$

end for; // Transformation of transitions //
for each place $p_i \in P$ **begin**
  $t_{in(p_i)} := {}^\bullet p_i$; // $|{}^\bullet p_i| = 1$ //
  $t_{out(p_i)} := p_i^\bullet$; // $|p_i^\bullet| = 1$ //
  remove place p_i and its corresponding arcs;
  $a := 0$;
  $s := 1$;
  Repeat

$$b := \left\lfloor \frac{M(p_i) + w(p_i) \cdot a}{v(p_i)} + 1 \right\rfloor \tag{6-2}$$

$$a := \left\lceil \frac{v(p_i) \cdot b - M(p_i)}{w(p_i)} \right\rceil \tag{6-3}$$

    if $a \le x_{in(p_i)}$ **then begin**
    Place p_i^s is added which connects transition $t_{in(p_i)}^a$ to transition $t_{out(p_i)}^{(b-1) \bmod x_{out(p_i)}+1}$.

$$\hat{M}(p_i^s) := \left\lfloor \frac{b-1}{x_{out(p_i)}} \right\rfloor \tag{6-4}$$

$$\delta(p_i^s) := \delta(t_{in(p_i)}) \tag{6-5}$$

    $s := s + 1$
    end if
  Until $a \ge x_{in(p_i)}$.
end for; // Transformation of places //

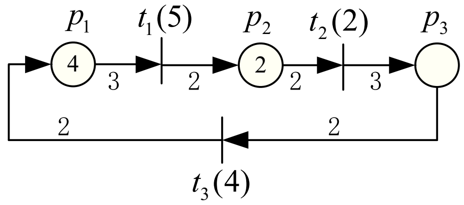

Fig. 6.1 The TWMG system $\langle N^\delta, M \rangle$ for Example 6.1.

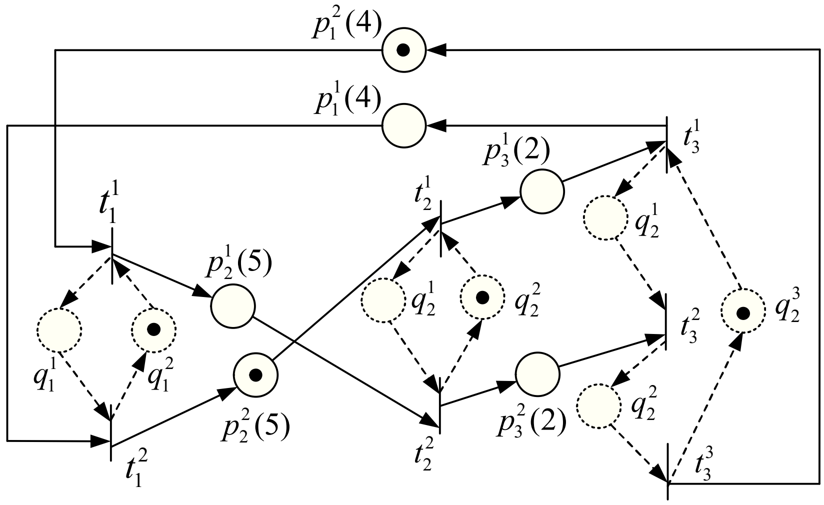

Fig. 6.2 The equivalent PTMG system $\langle \hat{N}^\delta, \hat{M} \rangle$ for Example 6.1.

(t_i^1,. . ., $t_i^{x_i}$)and x_i places (q_i^1,. . ., $q_i^{x_i}$). As a result, we replace transitions t_1, t_2, and t_3 by circuits $t_1^1 q_1^1 t_1^2 q_1^2$ t_i^1, $t_2^1 q_2^1 t_2^2 q_2^2$, and $t_3^1 q_3^1 t_3^2 q_3^2 t_3^3 q_3^3$, respectively. The markings of these places are specified in step 5.

Second, for each place p_j ($j = 1, 2, 3$), we replace it with a set of equivalent places. In particular, we replace place p_1, p_2, and p_3 by places p_1^1 and p_1^2, p_2^1, and p_2^2, and p_3^1 and p_3^2, respectively. The input and output arcs, the sojourn times, and the markings of these equivalent places are specified in steps 17-19. $\diamond$

6.3 Cycle Time Optimization Under Infinite Server Semantics: an Optimal Approach

In this subchapter, we present a formal approach to solve Problem (5-1) under infinite server semantics. First, we show that the cycle time optimization problem for PTMGs can be framed as an MILPP by modifying a known result to compute the cycle time of a PTMG whose initial marking is given. Second, we expose the conversion procedure from TWMGs to PTMGs. Although the exact structure of the equivalent PTMG depends on the initial marking of the TWMG (which is unknown when solving an optimization problem), we show that the number of possible equivalent structures is finite and periodic with the initial making on the TWMG. This means that in fact only a finite number of equivalent PTMG structures have to be considered. Finally, we propose an MILPP to solve the optimization problem for all the possible equivalent PTMGs.

6.3.1 Cycle Time Optimization of PTMGs

The cycle time optimization problem for a PTMG net $\hat{N}^\delta$ can be formulated as follows:

$$\begin{array}{l} \min\ \chi(\hat{M}) \\ \text{s.t.} \\ \quad \hat{y}^T \cdot \hat{M} \le R \end{array} \tag{6-6}$$

Proposition 6.1. Let $(\hat{M}^*, \beta^*, \alpha^*)$ be an optimal solution of the MILPP:

$$\begin{array}{l} \max\ \beta \\ \text{s.t.} \\ \begin{cases} \hat{C} \cdot \alpha + \hat{M} \ge D_p \cdot \hat{Post} \cdot v \cdot \beta, \\ \hat{y}^T \cdot \hat{M} \le R, \\ \hat{M} \in \mathbb{N}^{\hat{n}},\ \alpha \in \mathbb{R}^{\hat{m}},\ \beta \in \mathbb{R}^+, \end{cases} \end{array} \tag{6-7}$$

where v is the visit ratio vector which is equal to $\vec{\mathbf{1}}_{\hat{n}\times 1}$ and D_p is a $\hat{n} \times \hat{n}$ matrix such that $D_p(i,j) = \delta(p_i)$, when $i = j$ and otherwise $D_p(i,j) = 0$.

Then $\hat{M}^*$ is an optimal solution for problem (6-6) with an optimal cycle time $\chi(\hat{M}^*) = 1/\beta^*$.

Proof: In [95] it is shown that the cycle time of a PTMG system $\langle \hat{N}^\delta, \hat{M} \rangle$ can be directly obtained by solving the following LPP:

$$\begin{array}{l} \max\ \sigma^T \cdot D_p \cdot \hat{Post} \cdot v \\ \text{s.t.} \\ \quad \begin{cases} \sigma^T \cdot \hat{C} = \mathbf{0}, \\ \sigma^T \cdot \hat{M} = 1, \\ \sigma \ge \mathbf{0}, \end{cases} \end{array} \tag{6-8}$$

The dual problem of LPP (6-8) is

$$\begin{array}{l} \min\ \chi(\hat{M}) \\ \text{s.t.} \\ \quad \hat{C} \cdot z + \chi(\hat{M}) \cdot \hat{M} \ge D_p \cdot \hat{Post} \cdot v \end{array} \tag{6-9}$$

where the decision variables are $\chi(\hat{M}) \in \mathbb{R}^{+}$ and $z \in \mathbb{R}^{\hat{m}}$. Now let us consider problem (6-9). This problem can be easily converted into the problem of determining the maximal throughput (i.e., the inverse of cycle time) of the PTMG system, given the initial marking. To this end, we only need to replace $\chi(\hat{M})$ with its inverse $\beta = 1/\chi(\hat{M})$ and obtain the following LPP:

$$\begin{aligned} &\max\ \beta \\ &\text{s.t.} \\ &\ \hat{C} \cdot (\beta z) + \hat{M} \geq D_p \cdot \hat{Post} \cdot v \cdot \beta, \end{aligned} \tag{6-10}$$

where $\beta \in \mathbb{R}^{+}$, and $\beta z \in \mathbb{R}^{\hat{m}}$, i.e.,

$$\begin{aligned} &\max\ \beta \\ &\text{s.t.} \\ &\ \hat{C} \cdot \alpha + \hat{M} \geq D_p \cdot \hat{Post} \cdot v \cdot \beta, \end{aligned} \tag{6-11}$$

where $\alpha \in \mathbb{R}^{\hat{m}}$ and $\beta \in \mathbb{R}^{+}$ are the new decision variables. Finally, assuming that M is unknown but under a given constraint on the cost of resources, we have Eq. (6-7).

Now, we will prove that the optimal solution of Eq. (6-7) provides an optimal marking and the corresponding optimal throughput for problem (6-6) by contradiction. Let us assume that the optimal solution of Eq. (6-7) is ($\hat{M}_{opt}, \beta_{opt}, \alpha_{opt}$) and ($\hat{M}^*, \beta^*$) is the optimal solution of Eq. (6-6) with $\beta^* > \beta_{opt}$. By solving Eq. (6-11) with $\hat{M} = \hat{M}^*$, it is ensured that we can obtain an optimal throughput that is equal to β^*. Let α^* be the optimal value of vector α. This implies that ($\hat{M}^*, \beta^*, \alpha^*$) satisfies all constraints in Eq. (6-7) by assumption $\beta^* > \beta_{opt}$. This contradicts the assumption that ($\hat{M}_{opt}, \beta_{opt}, \alpha_{opt}$) is an optimal solution.

□

6.3.2 Transformation of the Cycle Time Optimization Problem of TWMGs into PTMGs

According to Algorithm 4, the net structure of the equivalent PTMG (i.e., the input and output arcs of equivalent places) is decided by the initial marking M_0 of the TWMG. Nevertheless, we will prove that this dependence is periodic in the following proposition.

Proposition 6.2. Given a TWMG N^δ with the minimal T-semiflow $x = (x_1, x_2, \ldots, x_m)^T$ and two different initial markings M_1 and M_2. Let $\langle \hat{N}_1^\delta, \hat{M}_1 \rangle$ (resp., $\langle \hat{N}_2^\delta, \hat{M}_2 \rangle$) be the equivalent PTMG system obtained by Algorithm 4 with input $\langle N^\delta, M_1 \rangle$ (resp., $\langle N^\delta, M_2 \rangle$).

For a place $p_i \in P$, if it satisfies

$$M_2(p_i) = M_1(p_i) + \xi \cdot v(p_i) \cdot x_{out(p_i)} \quad with\ \xi \in \mathbb{N}.$$

We deduce that the net structure corresponding to p_i in $\hat{N}_1^\delta$ and $\hat{N}_2^\delta$ is identical and the markings of the equivalent places p_i^s corresponding to p_i in Eq. (6-4) satisfy

$$\hat{M}_2(p_i^s) = \hat{M}_1(p_i^s) + \xi. \tag{6-12}$$

Proof: It is obvious that $M_1(p_i) + x_{in(p_i)} \cdot w(p_i) - x_{out(p_i)} \cdot v(p_i) = M_1(p_i)$, and thus we have $x_{in(p_i)} \cdot w(p_i) = x_{out(p_i)} \cdot v(p_i)$. For marking $M_1(p_i)$ of place p_i, it holds that:

$$\begin{cases} b_1 := \left\lfloor \frac{M_1(p_i)+w(p_i)\cdot a_1}{v(p_i)} + 1 \right\rfloor \\ a_1 := \left\lceil \frac{v(p_i)\cdot b_1 - M_1(p_i)}{w(p_i)} \right\rceil \end{cases} \tag{6-13}$$

and for marking $M_2(p_i)$ of place p_i,

$$\begin{cases} b_2 := \left\lfloor \frac{M_2(p_i)+w(p_i)\cdot a_2}{v(p_i)} + 1 \right\rfloor \\ a_2 := \left\lceil \frac{v(p_i)\cdot b_2 - M_2(p_i)}{w(p_i)} \right\rceil \end{cases} \tag{6-14}$$

since $M_2(p_i) = M_1(p_i) + \xi \cdot v(p_i) \cdot x_{out(p_i)}$

$$\begin{cases} b_2 := \left\lfloor \frac{M_1(p_i)+w(p_i)\cdot a_2}{v(p_i)} + 1 \right\rfloor + \xi \cdot x_{out(p_i)} \\ a_2 := \left\lceil \frac{v(p_i)\cdot b_2 - M_1(p_i)}{w(p_i)} \right\rceil - \xi \cdot x_{in(p_i)} \end{cases} \tag{6-15}$$

According to Algorithm 4, a_1 and a_2 are initialized to zero and after simplifying Eq. (6-15), we can obtain the following equation

$$\begin{cases} b_2 = b_1 + \xi \cdot x_{out(p_i)}, \\ a_2 = a_1, \\ (b_2 - 1) \bmod x_{out(p_i)} + 1 = (b_1 - 1) \bmod x_{out(p_i)} + 1 \end{cases} \tag{6-16}$$

Regarding to Eq. (6-16), it follows that the equivalent structures of $M_1(p_i)$ and $M_2(p_i)$ are identical while

$$\hat{M}_2(p_i^s) = \left\lfloor \frac{b_2 - 1}{x_{out(p_i)}} \right\rfloor = \hat{M}_1(p_i^s) + \xi.$$

□

Proposition 6.2 indicates that the equivalent structure corresponding to place p_i is periodic with respect to $M(p_i)$ and the period ϕ_i is equal to $v(p_i) \cdot x_{out(p_i)}$.

Example 6.2. Let us consider a simple TWMG model N^δ in Fig. 5.3 whose minimal T-semiflow is x=(2, 3)^T. Fig. 6.3 shows the equivalent PTMG systems $\langle \hat{N}^\delta, \hat{M} \rangle$ corresponding to different initial markings.

Transitions t_1 and t_2 are replaced by circuits $t_1^1 q_1^1 t_1^2 q_1^2$ and $t_2^1 q_2^1 t_2^2 q_2^2 t_2^3 q_2^3$, respectively.

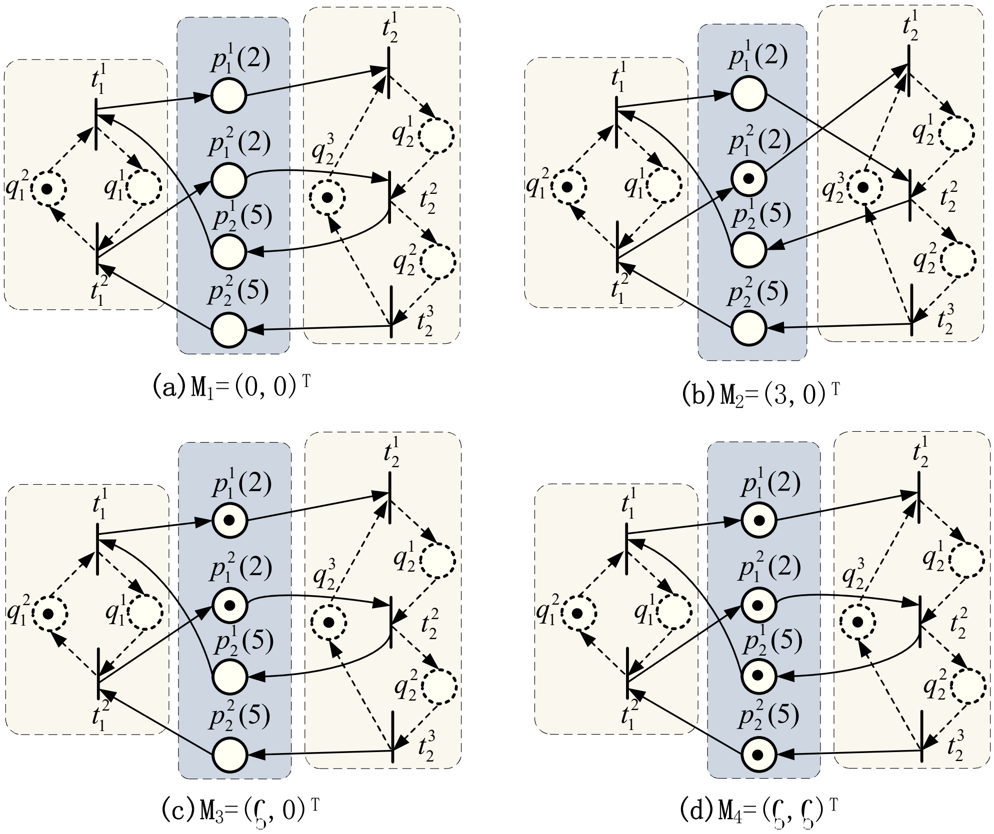

Fig. 6.3 The equivalent PTMG systems corresponding to different initial markings for Example 6.2.

Place p_1 (resp., p_2) is replaced by n_1=2 (resp., n_2=2) places p_1^1 and p_1^2 (resp., p_2^1 and p_2^2). For different initial markings, the structures of equivalent transitions (gray blocks) are always the same, while the structures and markings of equivalent places (blue blocks) may change.

Considering the equivalent PTMG system $\langle \hat{N}_2, \hat{M}_2 \rangle$ in Fig. 6.3(b), we denote the initial markings of equivalent places by $\mu(p_1^1) = \hat{M}(p_1^1) = 0$, $\mu(p_1^2) = \hat{M}(p_1^2) = 1$, $\mu(p_2^1) = \hat{M}(p_2^1) = 0$, and $\mu(p_2^2) = \hat{M}(p_2^2) = 0$.

The period of place p_1 (resp., p_2) is $\phi_1 = 6$ (resp., $\phi_2 = 6$). We can observe that the equivalent PTMG structures corresponding to M_1, M_3, and M_4 are the same as shown in Figs. 6.3(a), 6.3(c), and 6.3(d). $\diamond$

The number of possible equivalent PTMG structures is very big. Following the technique in Chapter 5 for single server semantics, we restrict our attention to the markings that belong to a restricted number of partitions which are guaranteed to find an optimal solution. In fact, to rule out the presence of useless tokens that do not contribute to the cycle time, we can assume that token content of each place p_i is a multiple of gcd_{p_i}. Thus the set of possible markings of place p_i can be partitioned into $\frac{\phi_i}{gcd_{p_i}}$ subsets such that

$$\bar{\mathcal{M}}_{p_i}^{k_i} = \{k_i \cdot gcd_{p_i} + \xi \cdot \phi_i | \xi \in \mathbb{N},\ k_i = 0, \ldots, \frac{\phi_i}{gcd_{p_i}} - 1\} \tag{6-17}$$

and all makings of p_i in the same partition $\bar{\mathcal{M}}_{p_i}^{k_i}$ correspond to the same equivalent structure.

Thus the set of possible markings of a TWMG is divided into Φ partitions:

$$\mathcal{M}_j = \bar{\mathcal{M}}_{p_1}^{k_{j,1}} \times \bar{\mathcal{M}}_{p_2}^{k_{j,2}} \times \ldots \times \bar{\mathcal{M}}_{p_n}^{k_{j,n}} \tag{6-18}$$

where

$$\Phi = \prod_{p_i \in P} \frac{\phi_i}{gcd_{p_i}}. \tag{6-19}$$

According to Proposition 5.6, the number of partitions in Eq. (6-19) can be further reduced to

$$\Phi''' = \prod_{p_j \in P \setminus P^*} \frac{\phi_j}{gcd_{p_j}} \times \prod_{p_i \in P^*} \frac{v(p_i)}{gcd_{p_i}}. \tag{6-20}$$

6.3.3 Optimal Approaches

According to Proposition 6.2, the equivalent structure of each place p_i is finite. We can compute the optimal solution of Problem (5-1) under infinite server semantics by considering all the possible equivalent PTMGs and solving MILPP (6-7) for each of them. However, there exist two critical issues that should be emphasized:

- The constraint on the cost of resources for a TWMG should be transformed into a new constraint for each equivalent PTMG.
- We have to add in Eq. (6-7) a series of constraints to guarantee the marking $\hat{M}$ that we find for a given net structure $\hat{N}^\delta$ is consistent with the marking M of N^δ that produces the structure $\hat{N}^\delta$.

For each place p_i with an initial marking

$$M(p_i) = k_{j,i} \cdot gcd_{p_i},\ k_{j,i} = 0, \ldots, \frac{\phi_i}{\text{gcd}_{p_i}} - 1, \tag{6-21}$$

we compute

- the equivalent structure of place p_i, i.e., places $p_i^1, \ldots, p_i^{n_i}$,
- the initial markings corresponding to Eq. (6-21), i.e., $\mu_j(p_i^1) = \hat{M}(p_i^1), \ldots, \mu_j(p_i^{n_i}) = \hat{M}(p_i^{n_i})$.

Thus for each partition $\mathcal{M}_j$ ($j = 1, \ldots, \Phi$), we can compute the equivalent PTMG system $\langle \hat{N}_j^{\delta} \hat{M}_j \rangle$.

Proposition 6.3. *For each partition* $\mathcal{M}_j$ ($j = 1, \ldots, \Phi'''$) *in Eq.* (6-18)*, let* ($\beta_j^*, M_j^*, \hat{M}_j^*, \hat{\alpha}_j^*, \xi_j^*$) *be an optimal solution of Eq.* (6-22)

$$
\begin{aligned}
&\max \beta_j \\
&\textit{s.t.} \\
&\begin{cases}
\hat{C}_j \cdot \alpha_j - D_p \cdot \hat{Post}_j \cdot v \cdot \beta_j + \hat{M}_j \geq 0, & (a) \\
y^T \cdot M_j \leq R, & (b) \\
M_j(p_i) = k_{j,i} \cdot gcd_{p_i} + \xi_{j,i} \cdot \phi_i, \ \forall p_i \in P, & (c) \\
\hat{M}_j(p_i^s) = \mu_j(p_i^s) + \xi_{j,i}, s = 1, \ldots, n_i, & (d) \\
\hat{M}_j(q_i^a) = 0, \ i = 1, \ldots, m, \ a = 1, \ldots, x_i - 1, & (e) \\
\hat{M}_j(q_i^{x_i}) = 1, \ i = 1, \ldots, m, & (f) \\
\xi_{j,i} \in \mathbb{N}, & (g)
\end{cases}
\end{aligned}
\tag{6-22}
$$

where $\beta_j \in \mathbb{R}_{\geq 0}$, $M_j \in \mathbb{N}^n$, $\hat{M}_j \in \mathbb{N}^{\hat{n}}$, $\hat{\alpha}_j \in \mathbb{R}^{\hat{m}}$. *Thus* M_j^* *is an optimal solution of Problem* (5-1) *under infinite server semantics that restricted to partition* $\mathcal{M}_j$.

Proof: The constraint (a) adopted from (6-11) can provide an optimal solution if $\hat{C}_j$, $\hat{Post}_j$ and D_p are given. The constraint (b) specifies that the weighted sum of tokens in places cannot exceed the upper bound on the cost of resources, and the constraint (c) specifies that feasible markings should be restricted to partition $\mathcal{M}_j$. The equivalent marking $\hat{M}_j$ is consistent with the marking M_j as ensured by constraints (d), (e) and (f). Thus ($\beta_j^*, M_j^*, \hat{M}_j^*, \hat{\alpha}_j^*, \xi_j^*$) is an optimal solution of Problem (5-1) under infinite server semantics that restricted to partition $\mathcal{M}_j$. □

Remark 6.1. Among all the Φ''' optimal solutions associated with each partition, we can obtain the maximal throughput and its corresponding marking, i.e., optimal solutions of Problem (5-1) under infinite server semantics. □

6.4 Cycle Time Optimization Under Infinite Server Semantics: Sub-optimal Approaches

The techniques introduced in Proposition 5.6 can significantly reduce the number of partitions of equivalent PTMGs. Nevertheless, we find that the number of partitions of equivalent structures still increases exponentially with the number of places. When this number becomes large, the efficiency of the optimal approach will be low and sometime it may not be possible to obtain an optimal solution because of the high computational cost. In this Section, we aim to reduce this cost while obtaining a near optimal solution.

6.4.1 Place Subset Allocation

From a theoretical point of view, it may be interesting to consider a subset of places to which resources are allocated instead of taking all the places into consideration, and we believe that in many cases this initial assignment may has a physical meaning that can lead to an optimal solution. As a result, the number of partitions of marking space can be significantly reduced since we consider $\frac{\phi_i}{gcd_{p_i}}$ partitions only for a subset of places p_i and one single partition for other places which are set to be empty. In the following, we will present an algorithm to select a subset of places to which resources should be allocated.

According to Theorem 3.1, a necessary condition to ensure the liveness of a TWMG is that all its elementary circuit are marked. In addition, if the weighted sum of tokens of each elementary circuit is greater than a constant value, then Proposition 3.2 provides a sufficient condition to ensure the liveness of a TWMG. Combining these two conditions, we select *at least one* place for each elementary circuit to which tokens are allocated.We assume the upper bound on the weighted sum of tokens is large enough such that the liveness of each elementary circuit can be guaranteed by putting enough tokens into the selected places.

We define a binary vector $\mathcal{I} \in \{0,1\}^n$, i.e.,

$$\mathcal{I} = (\mathcal{I}(1), \mathcal{I}(2), \cdots, \mathcal{I}(n))^T.$$

Tokens are initially allocated to the places p_j such that $\mathcal{I}(j)$ = 1 and we denote by the set of selected places as

$$P_r = \{p_j | \mathcal{I}(j) = 1\}.$$

To fulfill the requirement that each elementary circuit should be marked, we enforce the following constraint:

$$\sum_{p_j \in \gamma} \mathcal{I}(j) \geq 1, \ \forall \gamma \in \Gamma. \tag{6-23}$$

In addition, in order to reduce the number of partitions, we present three different approaches to compute the *place subset* P_r based on different objective functions.

PSA1: In this approach, we aim to minimize the number of places to which tokens should be added. Thus, the place subset P_r can be computed by solving the following problem:

$$\text{PSA1}: \quad \begin{cases} \min \vec{\mathbf{1}}_n^T \cdot \mathcal{I} \\ \text{s.t.} \\ \quad \sum_{p_j \in \gamma} \mathcal{I}(j) \geq 1, \ \forall \gamma \in \Gamma. \end{cases} \tag{6-24}$$

PSA2: In this approach, we aim to select the places which use the minimal cost of resources. First we define an n-dimensional vector

$$\boldsymbol{g}_d = (\text{gcd}_{p_1} \cdot y_1, \text{gcd}_{p_2} \cdot y_2, \cdots, \text{gcd}_{p_n} \cdot y_n)^T,$$

where y is the weight vector used in the criterion that represents the cost of the resources. Note that the number of useful tokens in place p_j should be a multiple of gcd_{p_j}. Therefore, the cost of resources used for place p_j should be a multiple of $\text{gcd}_{p_j} \cdot y_j$. Among all the places, we aim to choose the one whose value of $\text{gcd}_{p_j} \cdot y_j$ is the minimal. As a result, the place subset P_r can be computed by solving the following problem:

$$\text{PSA2}: \quad \begin{cases} \min \boldsymbol{g}_d^T \cdot \mathcal{I} \\ \text{s.t.} \\ \quad \sum_{p_j \in \gamma} \mathcal{I}(j) \geq 1, \ \forall \gamma \in \Gamma. \end{cases} \tag{6-25}$$

PSA3: In this approach, we aim to minimize the number of partitions and reduce the computational cost of optimal approach as much as possible. Thus, the place subset P_r can be computed by solving the following problem:

$$\text{PSA3}: \quad \begin{cases} \min \prod_{p_i \in P} \phi_i^{\mathcal{I}(i)} \\ \text{s.t.} \\ \quad \sum_{p_j \in \gamma} \mathcal{I}(j) \geq 1, \ \forall \gamma \in \Gamma. \end{cases} \tag{6-26}$$

Note that the objective function of PSA3 is equivalent to $\min \prod_{p_i \in P_r} \phi_i$. It may be unsolvable when the number of variables is large due to the fact that it is non-linear.

After we obtain the place subset P_r by solving the aforementioned approaches, we can look for possible suboptimal but computationally more efficient solutions of Problem (5-1) under infinite server semantics, as formalized in the following proposition.

Proposition 6.4. Let (M, β) be the optimal solution of the MILPP (6-22) by replacing constraint (c) with following constraints:

$$\begin{cases} M_j(p_i) = k_{j,i} \cdot \text{gcd}_{p_i} + \xi_{j,i} \cdot \phi_i, \ \forall p_i \in P_r, & (c1) \\ M_j(p_i) = 0, \ \forall p_i \notin P_r. & (c2) \end{cases} \tag{6-27}$$

where P_r is the place subset computed by any of the PSA approaches proposed above, i.e., PSA1, PSA2, or PSA3. If $\beta > 0$, then M is a (possibly sub-optimal) live solution for Problem (5-1) under infinite server semantics.

Proof: Constraint $c1$ is the same as constraint c in Eq. (6-22) and is only valid for the selected places. Constraint $c2$ ensures that the number of tokens in places that do not belong to the subset P_r should be zero which is used to reduce the number of partitions.

Thus, by solving Eq. (6-22) with new constraints $c1$ and $c2$ in Eq. (6-27), we obtain a live marking M if $\beta > 0$. □

As a result, the number of partitions for the PSA approaches is reduced to

$$\Phi^4 = \prod_{p_j \in P_r} \frac{\phi_j}{\text{gcd}_{p_j}}. \tag{6-28}$$

According to the results in Proposition 6.2, for a place p_i with zero token there may exist several markings that belong to the same partition. Thus, the solution obtained by MILPP (6-22) with new constraints in Eq. (6-27) may be improved by the following proposition.

Proposition 6.5. In MILPP (6-22), constraint $c2$ from Eq. (6-27) may be relaxed in

$$M_j(p_i) = \xi_{j,i} \cdot \phi_i,\ \xi_{j,i} \in \mathbb{N},\ \forall p_i \notin P_r.\ \ (c2')$$

The relaxed MILPP has a solution β greater than or equal to the original MILPP (6-22) and the same number of partitions Φ^4.

Proof: Given constraint $c2$ in Eq. (6-27), only one partition may correspond to places $p_i \notin P_r$ since they are marked with zero token. On the basis of Proposition 6.2, $M_j(p_i) = 0$ and $M_j(p_i) = \xi_{j,i} \cdot \phi_i$ belong to the same partition, while $M_j(p_i) = 0$ is a special case of $M_j(p_i) = \xi_{j,i} \cdot \phi_i$ when $\xi_{j,i} = 0$. As a consequence, the number of admissible markings is increased and the obtained throughput β of the PSA approaches may be improved by replacing constraint $c2$ in Eq. (6-27) with the more general constraint $c2'$, while the number of partitions Φ^4 remains the same. □

6.4.2 Throughput Upper Bound

It is shown in [79] that an upper bound of the throughput of a TWMG system $\langle N^\delta, M\rangle$ under infinite server semantics can be obtained by solving the following LPP:

$$\begin{aligned} & \max\ \beta' \\ & \text{s.t.} \\ & \quad C \cdot z + M - Pre \cdot \theta \cdot \beta' \geq \mathbf{0} \end{aligned} \tag{6-29}$$

where $\theta = (x_1 \cdot \delta(t_1), x_2 \cdot \delta(t_2), \ldots, x_m \cdot \delta(t_m))^T$ (recall x is the minimal T-semiflow of the TWMG and $\delta(t_i)$ is the delay time of t_i). The decision variables are $\beta' \in \mathbb{R}^+$ and $z \in \mathbb{R}^m$, and the optimal value of β' provides an upper bound of the throughput, i.e.,

$$\beta' \geq \beta. \tag{6-30}$$

As we discussed in the remark following Proposition 6.2, we can further refine the admissible domain by considering only markings whose number of tokens in any place p_i is a multiple of gcd_{p_i}.

Additionally, it may also happen that the marking obtained by Eq. (6-29) is a dead marking. Combining these results with Proposition 3.2, we present the following proposition.

Proposition 6.6. Let (M, β') be the optimal solution of the MILPP

$$\begin{aligned} &\max\ \beta' \\ &\text{s.t.} \\ &\begin{cases} y_\gamma^T \cdot M > W(M_D^\gamma), \forall \gamma \in \Gamma, \\ C \cdot z + M - Pre \cdot \theta \cdot \beta' \geq \mathbf{0}, \\ M(p_i) \text{ mod } \text{gcd}_{p_i} = 0,\ i = 1, \ldots, n, \\ y^T \cdot M \leq R. \end{cases} \end{aligned} \tag{6-31}$$

where y_γ denotes the P-semiflow associated with the elementary circuit γ. The decision variables are $\beta' \in \mathbb{R}_{\geq 0}$, $M \in \mathbb{N}^n$, and $z \in \mathbb{R}^m$. Then M is a sub-optimal live solution for Problem (5-1) under infinite server semantics and β' is an upper bound of the throughput that it produces.

Proof: According to Theorem 3.1, a TWMG is live iff each elementary circuit is live. The first constraint is a sufficient condition that ensures the liveness of a weighted elementary circuit according to Proposition 3.2. Thus, the marking M that we obtain by Eq. (6-31) will be a live marking. The second condition ensures that marking M is a sub-optimal solution with an upper bound of throughput β'. The number of tokens in place p_i should be a multiple of gcd_{p_i}, which is guaranteed by the third constraint. The fourth constraint is added to limit the cost of resources.

Summing up the above details, Eq. (6-31) gives a sub-optimal solution for Problem (5-1) under infinite server semantics. □

6.4.3 Computational Complexity Discussion

It is well known that ILPPs are NP-hard and it is common to characterize the computational burden by the number of variables and constraints [130]. The comparison of the number of variables and constraints for the proposed approaches is shown in Table 6.1. The three columns represent the numbers of variables, constraints, and MILPPs to solve, respectively. Note that in Table 6.1 $\hat{n}$, $\hat{m}$, $|P_r|$, $|\Gamma|$, and x represent the numbers of equivalent places of the PTMG, equivalent transitions of the PTMG, selected places for the PSA approach, the total elementary circuits, and the minimal T-semiflow of the TWMG, respectively.

For the optimal approach, the MILPP in Eq. (6-22) has $2n+\hat{n}+\hat{m}+1$ (β_j, M_j, $\hat{M}_j$, $\hat{\alpha}_j$, and $\xi_{j,i}$) variables and $2\hat{n}+n+\vec{\mathbf{1}}\cdot x+1$ constraints totally. The optimal approach requires solving Φ''' MILPPs in Eq. (6-22).

For the PSA approaches, the MILPP in Eq. (6-28) has $n+|P_r|+\hat{n}+\hat{m}+1$ (β_j, M_j, $\hat{M}_j$, $\hat{\alpha}_j$, and $\xi_{j,i}$) variables and $2\hat{n}+n+\vec{\mathbf{1}}\cdot x+1$ constraints totally. The PSA approach requires solving Φ^4 MILPPs in Eq. (6-28). We observe that the number of variables in Eq. (6-28) is smaller than that of Eq. (6-22) and the number of partition Φ^4 is also smaller than that of Eq. (6-22). In the worst case, $|P_r| = n$ and $\Phi^4 = \Phi'''$, i.e., the solution obtained from Eq. (6-24) contains all places. Then, the computational burden of the PSA approaches is the same with the optimal approach. However, in practical example, we find that the computational burden of the PSA approach is much smaller than that of the optimal approach.

For the Throughput Upper Bound (TUB) approach, the MILPP in Eq. (6-31) has $m+n+1$ (M, z, and β') variables and $n+m+|\Gamma|+1$ constraints totally. From a theoretical point of view, the total number of elementary circuits $|\Gamma|$ can grow exponentially with respect to the net size. However, we find that this number is quite reasonable in practice. In contrast to the optimal approach and the PSA approaches, the TUB approach requires to solve the MILPP (6-31) only once. In practical examples, the computational burden of the TUP approach is significantly smaller than both the optimal approach and the PSA approaches.

Table 6.1 Number of variables and constraints for the proposed approaches.

	Variables	Constraints	Number of ILPPs
Optimal approach	$2n+\hat{n}+\hat{m}+1$	$2\hat{n}+n+\vec{\mathbf{1}}\cdot x+1$	Φ'''
PSA approach	$n+\|P_r\|+\hat{n}+\hat{m}+1$	$2\hat{n}+n+\vec{\mathbf{1}}\cdot x+1$	Φ^4
TUB approach	$m+n+1$	$n+m+\|\Gamma\|+1$	1

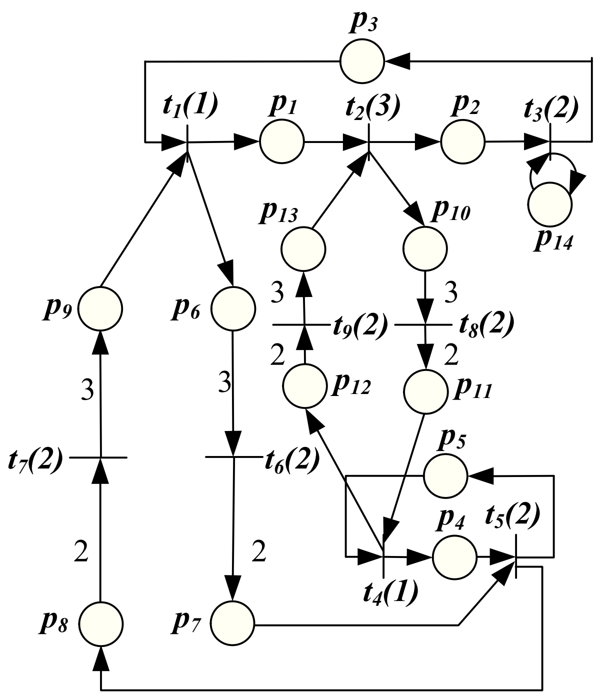

Fig. 6.4 The TWMG model N^{δ} for Example 6.3.

6.5 Experimental Study and Discussion

6.5.1 Application to a Flexible Manufacturing Systems

Example 6.3. The FMS combines cyclic assembly process, buffers, work in process, and batch operations. This system is composed of three machines $\mathcal{U}_1, \mathcal{U}_2$ and $\mathcal{U}_3$. It is cyclic and can manufacture two products, denoted by $\mathcal{R}_1$ and $\mathcal{R}_2$. The production ratios are 3/5 and 2/5 for $\mathcal{R}_1$ and $\mathcal{R}_2$, respectively. The production processes of these products are:

$$\begin{cases} \mathcal{R}_1: \ (\mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3) \\ \mathcal{R}_2: \ (\mathcal{U}_2, \mathcal{U}_1) \end{cases}$$

The TWMG model N^{δ} of the FMS is shown in Fig. 6.4 which is strongly connected and consist of seven elementary circuits: $\gamma_1 = p_1t_2p_2t_3p_3t_1$, $\gamma_2 = p_4t_5p_5t_4$, $\gamma_3 = p_{10}t_8p_{11}t_4p_{12}$ $t_9p_{13}t_2$, $\gamma_4 = p_6t_6p_7t_5p_8t_7p_9t_1$, $\gamma_5 = p_{14}t_3$, $\gamma_6 = p_2t_3p_3t_1p_6t_6p_7t_5p_5t_4p_{12}t_9p_{13}t_2$, and $\gamma_7 =$ $p_{10}t_8p_{11}t_4p_4t_5$ $p_8t_7p_9t_1p_1t_2$, where γ_1 and γ_2 are process circuits, γ_3, γ_4, and γ_5 are command circuits, and γ_6 and γ_7 are mixed circuits. The tokens in command circuits γ_3, γ_4, and γ_5 and process circuits γ_1 and γ_2 represent the servers and available pallets for products, respectively.

The minimal P-semiflows of γ_1, γ_2, γ_3, γ_4, γ_5, γ_6, and γ_7 are:

$$\begin{aligned}
y_1 &= (1,1,1,0,0,0,0,0,0,0,0,0,0,0)^T,\\
y_2 &= (0,0,0,1,1,0,0,0,0,0,0,0,0,0)^T,\\
y_3 &= (0,0,0,0,0,0,0,0,0,2,3,3,2,0)^T,\\
y_4 &= (0,0,0,0,0,2,3,3,2,0,0,0,0,0)^T,\\
y_5 &= (0,0,0,0,0,0,0,0,0,0,0,0,0,1)^T,\\
y_6 &= (0,2,2,0,3,2,3,0,0,0,0,3,2,0)^T,\\
y_7 &= (2,0,0,3,0,0,0,3,2,2,3,0,0,0)^T,
\end{aligned}$$

and the minimal T-semiflow is x=(3, 3, 3, 2, 2, 1, 1, 1, 1)^T. For each circuit we assume that $\lambda_\gamma = 1$ and the weight vector used in the criteria is $y = (3,3,3,4,4,4,6,6,4,4,6,6,4,1)^T$.

Assuming that the upper bound on the cost of resource R is equal to 100, Problem (5-1) under infinite server semantics can be immediately formulated as follows:

$$\begin{aligned}
&\min\ \chi(M)\\
&\text{s.t.}\\
&\begin{cases} y^T \cdot M \le 100,\\ y = (3,3,3,4,4,4,6,6,4,4,6,6,4,1)^T. \end{cases}
\end{aligned}$$

For the optimal approach, we have $\phi_1 = 3$, $\phi_2 = 3$, $\phi_3 = 3$, $\phi_4 = 2$, $\phi_5 = 2$, $\phi_6 = 3$, $\phi_7 = 2$, $\phi_8 = 2$, $\phi_9 = 3$, $\phi_{10} = 3$, $\phi_{11} = 2$, $\phi_{12} = 2$, $\phi_{13} = 3$, $\phi_{14} = 3$, and $\text{gcd}_{p_i} = 1$ ($i = 1, \ldots, 14$). Thus, the markings of the TWMG are partitioned into $\Phi = 139968$ subsets.

For the PSA approaches, we solve Eqs. (6-24), (6-25), and (6-26) by using Lingo and obtain the place subset P_r and the total number of partitions of equivalent PTMGs as shown in the following.

$$\begin{aligned}
&\text{PSA1}: \ P_r = \{p_1, p_4, p_7, p_{11}, p_{14}\},\ \Phi' = 72,\\
&\text{PSA2}: \ P_r = \{p_2, p_5, p_9, p_{13}, p_{14}\},\ \Phi' = 162,\\
&\text{PSA3}: \ P_r = \{p_2, p_4, p_8, p_{11}, p_{14}\},\ \Phi' = 72.
\end{aligned}$$

For the TUB approach, we can obtain a sub-optimal solution by solving the following problem:

$\max\ \beta'$

s.t.

$M(1) + M(2) + M(3) > 0,$

$M(4) + M(5) > 0,$

$2M(10) + 3M(11) + 3M(12) + 2M(13) > 7,$

$2M(6) + 3M(7) + 3M(8) + 2M(9) > 7,$

$M(14) > 0,$

$2M(2) + 2M(3) + 3M(5) + 2M(6) + 3M(7) + 3M(12) + 2M(13) > 7,$

$2M(1) + 3M(4) + 3M(8) + 2M(9) + 2M(10) + 3M(11) > 7,$

Table 6.2 Simulation results for Example 6.3.

Approach X	Nb. of partitions	Obtained marking M	β_{X}	β'	G_X	G'_X	CPU time [s]
Optimal	139968	$(6,0,0,0,2,3,0,$ $4,0,0,0,6,0,2)^T$	0.31		0%	13.9%	57840
PSA1	72	$(0,0,6,0,2,0,0$ $6,0,0,2,4,0,2)^T$	0.31		0%	13.9%	23
PSA2	162	$(0,6,0,0,2,0,0$ $6,0,0,2,4,0,2)^T$	0.31	0.36	0%	13.9%	48
PSA3	72	$(0,0,6,0,2,0,0$ $6,0,0,2,4,0,2)^T$	0.31		0%	13.9%	23
TUB	N/A	$(0,0,7,3,0,5,1,$ $0,1,8,0,0,1,1)^T$	0.17		45.2%	52.8%	4

$C \cdot z + M - Pre \cdot \theta \cdot \beta' \geq \mathbf{0}$,
$M(p_i)$ mod $\text{gcd}_{p_i} = 0,\ i = 1, \ldots, n$,
$y^T \cdot M \leq 100$.

The experiment results are carried out on a PC with a Pentium Dual-Core CPU 3.0 GHz using MATLAB with YALMIP [129]. For a given approach X (where $X \in \{$Optimal, PSA1, PSA2, PSA3, TUB $\}$), we define the *optimality gap*

$$G_X = (\beta_{opt} - \beta_{\mathrm{X}})/\beta_{opt} \tag{6-32}$$

the difference in $\%$ between the optimal throughput β_{opt} and the throughput computed with approach X, and the *optimality gap upper bound*

$$G'_X = (\beta' - \beta_X)/\beta' \tag{6-33}$$

the difference in $\%$ between the upper bound on the throughput β' computed with the TUB approach and the throughput computed with approach X.

In Table 6.2, we show the tested approach, the number of partitions Φ' (resp., Φ'') that must be considered for the optimal approach (resp., PSA approaches), the obtained marking M, the throughput β_X computed with approach X, the throughput upper bound β' obtained with the TUB approach, the optimality gap G_X, the optimality gap upper bound G'_X, and the CPU time for each approach.

In this example, we solve PSA1, PSA2, and PSA3 approaches by MILPP (6-22) with new constraints $c1$ in Eq. (6-27) and $c2'$ in Proposition 6.5 and the obtained solutions are optimal. Nevertheless, this result does not hold in general, i.e., the PSA approaches cannot

Table 6.3 Used resources for the obtained solutions.

	Cost for pallets	Cost for servers
Optimal	26	74
PSA1	26	74
PSA2	26	74
PSA3	26	74
TUB	33	67

always provide an optimal solution. The number of partitions of PSA1 and PSA3 are smaller than that of PSA2. Due to the reduced number of partitions, the CPU times required by the PSA approaches are much smaller than that of the optimal approach. The solutions obtained by TUB are good candidates and the CPU time is the minimal one. The cost for pallets and servers of solutions obtained by the proposed approaches are reported in Table 6.3. It is shown that the resource distributions for the solutions obtained by the optimal approach and the PSA approaches are identical, while that of the TUB approach spends more cost of resources on pallets and less cost of resources on servers.

We mention that the computational cost of the optimal approach and the PSA approach can be influenced by the arcs of the TWMG model tremendously. For example, if we change the production ratios for $\mathcal{R}_1$ and $\mathcal{R}_2$ to 2/3 and 1/3, the arcs of the command circuits will be changed accordingly, i.e., $Pre(p_6, t_6) = 2$, $Post(p_7, t_6) = 1$, $Pre(p_8, t_7) = 1$, $Post(p_9, t_7) = 2$, $Pre(p_{10}, t_8) = 2$, $Post(p_{11}, t_8) = 1$, $Pre(p_{12}, t_9) = 1$, and $Post(p_{13}, t_9) = 2$. Thus, the number of partitions for optimal approach and PSA approach are 64 and 8, respectively. Nevertheless, if we change the production ratios for $\mathcal{R}_1$ and $\mathcal{R}_2$ to $7/10$ and $3/10$, these numbers will increase to 6.004e+9 and 7203, respectively.

6.5.2 More Cases Study

To better investigate the efficiency of the proposed approaches and the sub-optimality gap for solutions obtained by the PSA and the TUB approaches, we analyzed some examples taken from literature. Case 1 is an assembly line taken from [64]. Case 2 is a jobshop taken from [95] that contains four process circuits. Case 3 is a slight modification of the jobshop in [95] with the addition of one process circuit.

In Table 6.4, we show for each considered instance the number of places and transitions, the upper bound on the cost of resources R, the tested approach, the number of partitions Φ' (resp., Φ^4) for the optimal approach (resp., PSA approaches), the throughput β_X computed with approach X, the throughput upper bound β' obtained with the TUB approach, the optimality gap G_X as defined in Eq. (6-32), the optimality gap upper bound G'_X as defined

Table 6.4 Simulation results for different instances.

	$\lvert P\rvert$	$\lvert T\rvert$	R	Approach X	Nb. of partitions	β_X	β'	G_X	G'_X	CPU time [s]
Case1	8	5	1000	Optimal	216	0.23	0.23	0%	0%	70
				PSA1	1	0.23		0%	0%	4
				PSA2	36	0.23		0%	0%	12
				PSA3	1	0.23		0%	0%	4
				TUB	N/A	0.21		8.7%	8.7%	3
Case2	24	12	1000	Optimal	1.00e+11	o.o.t	0.38	N/A	N/A	o.o.t
				PSA1	3456	0.31		N/A	18.4%	1096
				PSA2	10368	0.27		N/A	28.9%	4981
				PSA3	3456	0.22		N/A	42.1%	1104
				TUB	N/A	0.36		N/A	5.3%	7
Case3	30	15	1000	Optimal	8.67e+15	o.o.t	0.32	N/A	N/A	o.o.t
				PSA1	6912	0.17		N/A	46.9%	3219
				PSA2	62208	0.24		N/A	25%	123057
				PSA3	6912	0.2		N/A	37.5%	3132
				TUB	N/A	0.29		N/A	9.4%	7

in Eq. (6-33), and the CPU time for each approach.

The simulation results show the tradeoff between computational cost and quality of the solution. Note that "o.o.t" (out of time) in Table 6.4 means that the solution cannot be found within 48 hours. The computational cost of the optimal solution can grow exponentially as the net size increases. For Case 1, the PSA approaches can provide an optimal throughput which in this case coincides with the upper bound on the throughput. Actually, if we find a solution whose throughput is equal to the upper bound on the throughput by using PSA approaches, we can deduce that this solution is also optimal. For Cases 2 and 3, the number of partitions required by the optimal approach is so large that we cannot obtain a solution within a reasonable computation time. The solutions obtained by the TUB approach for Cases 2 and 3 are better than those of the PSA approach. It is not obvious that which approach is the best among the three PSA approaches, i.e., PSA1, PSA2, and PSA3. We observe that the upper bound gap of solutions obtained by the TUB approach for Cases 1, 2, and 3 are quite small, which means that these solutions are very close to be optimal.

As an advancement, we can say, from the set of examples optimized, that the sub-optimal approaches can obtain high quality approximate solutions within quite reasonable computational effort. Moreover, with respect to the optimal approach, the computational time of the sub-optimal approaches is quite small. In practice, reduction of the computational cost is very important and necessary. Thus, the presented sub-optimal approaches try to cope with this requirement.

6.6 Conclusion

In this chapter, we address the cycle time optimization for TWMGs under infinite server semantics, which is a more general case than the one we discussed in Chapter 5. We aim to find a proper schedule such that the weighted sum of tokens in places is less than or equal to a given value and the cycle time is minimized. We show that the performance optimization for a TWMG can be transformed into the performance optimization for a finite family of PTMGs under the condition that the initial marking of the TWMG is not given. An optimal approach is developed to solve the optimization problem. Nevertheless, the computational cost of the optimal approach can grow exponentially as the net size increases. Then, we propose three sub-optimal solutions to reduce the computational burden which considers a subset of places. Finally, an MILPP based on the upper bound of the throughput is formulated, which is practically efficient.

The results presented in this chapter have also been published in:

Z. He, Z. W. Li, and A. Giua, "Cycle time optimization for deterministic timed weighted marked graphs under infinite server semantics," In *Proceedings of the 55th iEEE International Conference on Decision and Control*, (CDC'16), 2016: 3942-3947.

Chapter 7 Conclusions and Future Research

This dissertation deals with performance optimization of automated manufacturing systems in the framework of timed Petri nets. This chapter summarizes the main contribution of this dissertation and introduces future research on timed Petri nets.

7.1 Contributions

- In Chapter 3, we study the marking optimization of deterministic TWMGs under single server semantics, which consists in finding an initial resource assignment to minimize the cost of resources while the system's throughput is less than or equal to a given value. The existing results fail to provide practically effective and computationally efficient methods to analyze and solve the problems in such systems. To this end, an efficient heuristic method is proposed to reduce the computational cost. We take the advantages of the net structure characteristics of a TWMG and utilize related knowledge of liveness of a TWMG to select a proper initial marking. Next, based on simulation a heuristic algorithm used to increase the system's throughput by adding tokens to some places is developed. Finally, a technique to improve the quality of the obtained solution by taking the advantages of the previous works is proposed. Several simulation studies show that the effectiveness of the proposed approach is significantly faster than existing ones.

- In Chapter 4, we investigate the marking optimization problem of deterministic TWMGs under infinite server semantic, which is more general than the one in Chapter 3. We propose two new heuristic approaches to obtain a near optimal solution. The proposed algorithms can also be applied to the marking optimization for deterministic TWMGs under single server semantics by adding to each transition a self-loop place with one token.

- In Chapter 5, the cycle time optimization of deterministic TWMGs under single server semantics is originally proposed, which is a dual problem of marking optimization problem. It consists in finding an initial resource assignment to maximize the system's throughput while the cost of resources is less than or equal to a given value. Periodicity of transformation of TWMGs into equivalent TMGs is formalized and the initial marking of a TWMG is partitioned into several subsets with regard to the periodicity.

By this transformation method a practically efficient algorithm is proposed to solve the optimization problem based on solving a series of MILPP. Finally this approach is further extended to a generalized optimization problem which maximizes the system's throughput and minimizes the cost of the resources.

- In Chapter 6, we study the cycle time optimization of deterministic TWMGs under infinite server semantic, which is more general than the one in Chapter 5. We consider the transformation of a given TWMG into an equivalent TMG proposed by Nakamura and Silva [95] and prove that this transformation is periodical with regard to the initial marking. This allow us to transform a TWMG into a finite family of equivalent TMGs, each one valid for a partition of set of initial markings. Then, we present an MILPP to solve the optimization problem that requires finding an optimal allocation for the equivalent TMG under the constraint that the initial marking belongs to a particular partition. This procedure, that can guarantee the convergence to the optimum, has a high computational complexity due to the fact that the number of partitions can increase exponentially with the number of places. Finally, two sub-optimal approaches without enumerating the entire partitions are proposed in order to reduce the computational complexity.

7.2 Future Work

Despite some heuristic algorithms are proposed to solve the marking optimization of TWMG models in this thesis, these heuristic approaches can only provide near optimal solutions. The problem of finding an optimal solution for this problem is still open. Our future works aim to develop an analytical method to solve this problem and find an optimal solution. The second perspective is to extend these results to continuous WMGs. To the best of our knowledge, the optimization problems of continuous WMG has not been addressed in the literature.

Considering the cycle time optimization problem of TWMGs studied in this dissertation, the proposed MILPP approach can guarantee the convergence to the optimum based on transforming a TWMG into a set of equivalent TMGs. Nevertheless, this transformation technique can lead to a huge number of equivalent TMGs in general. As a consequence, the time consumption for solving the MILPPs is very high. Our future work includes providing an optimal approach that is directly applicable to TWMGs, i.e., without the transformation procedure.

Another restriction of both TMGs and TWMGs is the fact that they cannot describe systems with choice, i.e., a condition where several future evolutions are possible but in conflict among them. For this reason, we plan to extend the considered modelling framework by assuming that choices are possible and must be resolved with a stationary routing, that assigns resources to conflicting processes with a preassigned ratio. The routing parameters will be additional decision variable of our optimization problem.

Reference

[1] VISWANADHAM N, NARAHARI Y. Performance modeling of automated manufacturing systems[M]. [S.l.] : Prentice Hall Englewood Cliffs, NJ, 1992.

[2] MURATA G T. Petri nets: properties, analysis and applications[J]. Proceedings of the IEEE, 1989, 77(4) : 541 – 580.

[3] MAGOTT J. Performance evaluation of concurrent systems using Petri nets[J]. Information Processing Letters, 1984, 18(1) : 7 – 13.

[4] CASSANDRAS C G, LAFORTUNE S. Introduction to discrete event systems[M]. [S.l.] : Springer, 2008.

[5] GIUA A, DICESARE F. Blocking and controllability of Petri nets in supervisory control[J]. IEEE Transactions on Automatic Control, 1994, 39(4) : 818 – 823.

[6] CHEN Y F, LI Z W, BARKAOUI K, et al. On the enforcement of a class of nonlinear constraints on Petri nets[J]. Automatica, 2015, 55 : 116 – 124.

[7] IORDACHE M V, ANTSAKLIS P J. Supervision based on place invariants: A survey[J]. Discrete Event Dynamic Systems, 2006, 16(4) : 4451 – 492.

[8] IORDACHE M V, ANTSAKLIS P J. Petri net supervisors for disjunctive constraints[C] // In Proceedings of the 26th American Control Conference. 2007 : 4951 – 4956.

[9] IORDACHE M V, ANTSAKLIS P J. Decentralized control of Petri nets with constraint transformation[J]. IEEE Transactions on Automatic Control, 2006, 51(2) : 376 – 381.

[10] IORDACHE M V, WU P, ZHU F, et al. Efficient design of Petri net supervisors with disjunctive specifications[C] // In Proceedings of the IEEE International Conference on Automation Science and Engineering. 2013 : 936 – 941.

[11] LUO J L, WU W M, SU H Y, et al. Supervisor synthesis for enforcing GMECs on a controlled Petri net[C] // In Proceedings of the 25th American Control Conference. 2006 : 4165 – 4170.

[12] LUO J L, WANG S G. Supervisor synthesis for enforcing a disjunction of GMECs on controlled Petri nets[C] // In Proceedings of the IEEE International Conference on Mechatronics and Automation. 2007 : 294 – 298.

[13] BRANDIN B A, WONHAM W M. Supervisory control of timed discrete-event systems[J]. IEEE Transactions on Automatic Control, 1994, 39(2) : 329 – 342.

[14] MOODY J, ANTSAKLIS P J. Supervisory control of discrete event systems using Petri nets[M]. [S.l.] : Springer Science & Business Media, 2012.

[15] LUO J L, NONAMI K. Approach for transforming linear constraints on Petri nets[J]. IEEE Trans-

actions on Automatic Control, 2011, 56(12) : 2751 – 2765.

[16] LUO J L, SHAO H, NONAMI K, et al. Maximally permissive supervisor synthesis based on a new constraint transformation method[J]. Automatica, 2012, 48(6) : 1097 – 1101.

[17] BASILE F, GIUA A, SEATZU C. Some new results on supervisory control of Petri nets with decentralized monitor places[C] // In Proceedings of the 17th IFAC World Congress. 2008 : 531 – 536.

[18] BASILE F, CORDONE R, PIRODDI L. Integrated design of optimal supervisors for the enforcement of static and behavioral specifications in Petri net models[J]. Automatica, 2013, 49(11) : 3432 – 3439.

[19] BASILE F, CORDONE R, PIRODDI L. A branch and bound approach for the design of decentralized supervisors in Petri net models[J]. Automatica, 2015, 52 : 322 – 333.

[20] BARKAOUI K, CHAOUI A, ZOUARI B. Supervisory control of discrete event systems based on structure theory of Petri nets[C] // In Proceedings of the International Conference on Systems, Man, and Cybernetics. 1997 : 3750 – 3755.

[21] POCCI M, DEMONGODIN I, GIAMBIASI N, et al. Synchronizing sequences on a class of unbounded systems using synchronized Petri nets[J]. Discrete Event Dynamic Systems, 2016, 26(1) : 85 – 108.

[22] SCHUPPEN J H V, SILVA M, SEATZU C. Control of discrete-event systems-automata and Petri net perspectives[J]. Lecture Notes in Control and Information Science, 2012 : 319 – 340.

[23] RAMCHANDANI C. Analysis of asynchronous concurrent systems by timed Petri nets[J], 1974.

[24] CHRZASTOWSKI-WACHTEL P, RACZUNAS M. Liveness of weighted circuits and the Diophantine problem of Frobenius[C] // In Proceedings of the International Symposium on Fundamentals of Computation Theory. 1993 : 171 – 180.

[25] MARCHETTI O, MUNIER-KORDON A. A sufficient condition for the liveness of weighted event graphs[J]. European Journal of Operational Research, 2009, 197(2) : 532 – 540.

[26] MARCHETTI O, MUNIER-KORDON A. Complexity results for weighted timed event graphs[J]. Discrete Optimization, 2010, 7(3) : 166 – 180.

[27] TERUEL E, CHRZASTOWSKI-WACHTEL P, COLOM J M, et al. On weighted T-systems[C] // In Proceedings of the International Conference on Application and Theory of Petri Nets. 1992 : 348 – 367.

[28] CHRZASTOWSKI-WACHTEL P, RACZUNAS M. Orbits, half-frozen tokens and the liveness of weighted circuits[J], 1995 : 116 – 128.

[29] POPOVA-ZEUGMANN L. Time and Petri nets[M]. [S.l.] : Springer, 2013 : 31 – 137.

[30] LI Z W, ZHOU M C. Elementary siphons of Petri nets and their application to deadlock prevention

in flexible manufacturing systems[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part A, 2004, 34(1) : 38 – 51.

[31] EZPELETA J, COLOM J M, MARTINEZ J. A Petri net based deadlock prevention policy for flexible manufacturing systems[J]. IEEE Transactions on Robotics and Automation, 1995, 11(2) : 173 – 184.

[32] LI Z W, ZHOU M C. Clarifications on the definitions of elementary siphons in Petri nets[J]. IEEE Transactions on Systems, Man and Cybernetics, Part A, 2006, 36(6) : 1227 – 1229.

[33] ZHAO M, HOU Y. An iterative method for synthesizing non-blocking supervisors for a class of generalized Petri nets using mathematical programming[J]. Discrete Event Dynamic Systems, 2011, 23(1) : 3 – 26.

[34] LI Z W, ZHAO M. On controllability of dependent siphons for deadlock prevention in generalized Petri nets[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part A, 2008, 38(2) : 369 – 384.

[35] ZHOU M C. Petri nets in flexible and agile automation[M]. [S.l.] : Springer, 1995.

[36] LI Z W, ZHOU M C. On siphon computation for deadlock control in a class of Petri nets[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part A, 2008, 38(3) : 667 – 679.

[37] LI Z W, ZHOU M C. Control of elementary and dependent siphons in Petri nets and their application[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part A, 2008, 38(1) : 133 – 148.

[38] CORDONE R, NAZEEM A, PIRODDI L, et al. Designing optimal deadlock avoidance policies for sequential resource allocation systems through classification theory: existence results and customized algorithms[J]. IEEE Transactions on Automatic Control, 2013, 58(11) : 2772 – 2787.

[39] LI Z W, ZHOU M C. Deadlock resolution in automated manufacturing systems: A novel Petri net approach[M]. London : Springer, 2009.

[40] LI Z W, WU N Q, ZHOU M C. Deadlock control of automated manufacturing systems based on Petri nets – A literature review[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part C, 2012, 42(4) : 437 – 462.

[41] CABASINO M, GIUA A, SEATZU. C. Fault detection for discrete event systems using Petri nets with unobservable transitions[J]. Automatica, 2010, 46(9) : 1531 – 1539.

[42] CABASINO M, GIUA A, POCCI M, et al. Discrete event diagnosis using labeled Petri nets: An application to manufacturing systems[J]. Control Engineering Practice, 2011, 19(9) : 989 – 1001.

[43] BOEL R K, JIROVEANU G. Distributed contextual diagnosis for very large systems[C] // In Proceedings of the 7th International Workshop on Discrete Event Systems. 2004 : 333 – 344.

[44] BASILE F, CHIACCHIO P, De Tommasi G. An efficient approach for online diagnosis of discrete event systems[J]. IEEE Transactions on Automatic Control, 2009, 54(4) : 748 – 759.

[45] BOEL R K, Van Schuppen J H. Decentralized failure diagnosis for discrete-event systems with costly communication between diagnosers[C] // In Proceedings of the 6th International Workshop on Discrete Event Systems. 2002 : 175 – 181.

[46] RAMIREZ-TREVINO A, RUIZ-BELTRAN E, RIVERA-RANGEL I, et al. Online fault diagnosis of discrete event systems: A Petri net-based approach[J]. IEEE Transactions on Automation Science and Engineering, 2007, 4(1) : 31 – 39.

[47] LEFEBVRE D, DELHERM C. Diagnosis of DES with Petri net models[J]. IEEE Transactions on Automation Science and Engineering, 2007, 4(1) : 114 – 118.

[48] BASILE F. Overview of fault diagnosis methods based on petri net models[C] // In Proceedings of the European Control Conference. 2014 : 2636 – 2642.

[49] USHIO T, ONISHI I, OKUDA K. Fault detection based on Petri net models with faulty behaviors[C] // In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics. 1998 : 113 – 118.

[50] BASILE F, CABASINO M P, SEATZU C. State estimation and fault diagnosis of labeled time petri net systems with unobservable transitions[J]. IEEE Transactions on Automatic Control, 2015, 60(4) : 997 – 1009.

[51] WANG X, MAHULEA C, SILVA M. Diagnosis of time Petri nets using fault diagnosis graph[J]. IEEE Transactions on Automatic Control, 2015, 60(9) : 2321 – 2335.

[52] MAHULEA C, SEATZU C, CABASINO M P, et al. Fault diagnosis of discrete-event systems using continuous Petri nets[J]. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 2012, 42(4) : 970 – 984.

[53] DOTOLI M, FANTI M P, MANGINI A M, et al. On-line fault detection in discrete event systems by Petri nets and integer linear programming[J]. Automatica, 2009, 45(11) : 2665 – 2672.

[54] FANTI M P, MANGINI A M, UKOVICH W. Fault detection by labeled Petri nets and time constraints[C] // In Proceedings of the 3rd International Workshop on the Dependable Control of Discrete Systems. 2011 : 168 – 173.

[55] SIFAKIS J. Performance evaluation of systems using nets[C] // In Proceedings of the Advanced Course on General Net Theory of Processes and Systems. 1980 : 307 – 319.

[56] GU Z, SHIN K G. An integrated approach to modeling and analysis of embedded real-time systems based on timed petri nets[C] // In Proceedings of the 23rd International Conference on Distributed Computing Systems. 2003 : 350 – 359.

[57] BENAZOUZ M, MARCHETTI O, MUNIER-KORDON A, et al. A polynomial algorithm for the computation of buffer capacities with throughput constraint for embedded system design[C] // In Proceedings of the International Conference on Computers and Industrial Engineering. 2009 :

690 – 695.

[58] BENAZOUZ M, MARCHETTI O, MUNIER-KORDON A, et al. A new approach for minimizing buffer capacities with throughput constraint for embedded system design[C] // In Proceedings of the International Conference on Computer Systems and Applications. 2010 : 1 – 8.

[59] HUJSA T, DELOSME J-M, MUNIER-KORDON A. Polynomial sufficient conditions of well-behavedness and home markings in subclasses of weighted Petri nets[J]. ACM Transactions on Embedded Computing Systems, 2014, 13(4) : 141.

[60] AMORIM L, BARRETO R, MACIEL P, et al. A methodology for software synthesis of embedded real-time systems based on TPN and LSC[C] // In Proceedings of the International Conference on Embedded Software and Systems. 2005 : 50 – 62.

[61] HOU G, CHANG J, SHOU K, et al. Embedded system modeling and verification based on deterministic and stochastic Petri net[J]. Journal of Computational Information Systems, 2014, 10(12) : 5051 – 5058.

[62] PROTH J M, SAUER N, XIE X L. Optimization of the number of transportation devices in a flexible manufacturing system using event graphs[J]. IEEE Transactions on Industrial Electronics, 1997, 44(3) : 298 – 306.

[63] LAFTIT S, PROTH J M, XIE X L. Optimisation of invariant criteria for event graphs[J]. IEEE Transactions on Automatic Control, 1992, 37(5) : 547 – 555.

[64] BENABID-NAJJAR A, HANEN C, MARCHETTI O, et al. Periodic schedules for bounded timed weighted event graphs[J]. IEEE Transactions on Automatic Control, 2012, 57(5) : 1222 – 1232.

[65] MERLIN P. A methodology for the design and implementation of communication protocols[J]. IEEE Transactions on Communications, 1976, 24(6) : 614 – 621.

[66] LOPEZ-MELLADO E. Simulation of timed Petri net models[C] // In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics. 1995 : 2270 – 2273.

[67] CAMPOS J, COLOM J M, JUNGNITZ H, et al. Approximate throughput computation of stochastic marked graphs[J]. IEEE Transactions on Software Engineering, 1994, 20(7) : 526 – 535.

[68] CHIOLA G, ANGLANO C, CAMPOS J, et al. Operational analysis of timed Petri nets and application to the computation of performance bounds[C] // In Proceedings of the 5th International Workshop on Petri Nets and Performance Models. 1993 : 128 – 137.

[69] CAMPOS J, CHIOLA G, COLOM J M, et al. Properties and performance bounds for timed marked graphs[J]. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 1992, 39(5) : 386 – 401.

[70] CHEN Y L, HSU P Y, CHANG Y B. A Petri net approach to support resource assignment in project management[J]. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems

and Humans, 2008, 38(3) : 564 – 574.

[71] LÓPEZ-MELLADO E. Analysis of discrete event systems by simulation of timed Petri net models[J]. Mathematics and Computers in Simulation, 2002, 61(1) : 53 – 59.

[72] ZHANG H, LIU F, YANG M, et al. Simulation of time Petri nets[C] // In Proceedings of the 4th International Conference on System Science, Engineering Design and Manufacturing Informatization. 2013 : 3 – 6.

[73] SAUER N. Marking optimization of weighted marked graphs[J]. Discrete Event Dynamic Systems, 2003, 13(3) : 245 – 262.

[74] TOURSI L, SAUER N. Branch and bound approach for marking optimization problem of weighted marked graphs[C] // In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics. 2004 : 1777 – 1782.

[75] HUANG Z, WU Z. A deadlock-free scheduling method for automated manufacturing systems using dynamic-edge graph with tokens[C] // In Proceedings of the IEEE International Conference on Control Applications. 2004 : 1398 – 1403.

[76] ZUBEREK W. Event-driven simulation of timed Petri net models[C] // In Proceedings of the 33rd Annual Simulation Symposium. 2000 : 91 – 98.

[77] MARSAN M A, BALBO G, CONTE G, et al. Modelling with generalized stochastic Petri nets[M]. [S.l.] : John Wiley & Sons, Inc., 1994.

[78] HILLION H P, PROTH J M. Performance evaluation of job-shop systems using timed event-graphs[J]. IEEE Transactions on Automatic Control, 1989, 34(1) : 3 – 9.

[79] CAMPOS J, CHIOLA G, SILVA M. Ergodicity and throughput bounds of Petri nets with unique consistent firing count vector[J]. IEEE Transactions on Software Engineering, 1991, 17(2) : 117 – 125.

[80] BACCELLI F, COHEN G, OLSDER G, et al. Synchronization and linearity: an algebra for discrete event systems[J]. Journalof the operational Research Society, 1994, 45(1) : 118 – 119.

[81] COHEN G, GAUBERT S, QUADRAT J P. Timed-event graphs with multipliers and homogeneous min-plus systems[J]. IEEE Transactions on Automatic Control, 1998, 43(9) : 1296 – 1302.

[82] KOMENDA J, LAHAYE S, BOIMOND J-L. Determinization of timed Petri nets behaviors[J]. Discrete Event Dynamic Systems, 2016, 26(3) : 413 – 437.

[83] DECLERCK P. Compromise approach for predictive control of Timed Event Graphs with specifications defined by P-time Event Graphs[J]. Discrete Event Dynamic Systems, 2016, 26(4) : 611 – 632.

[84] BOUSSAHEL W M, AMARI S, KARA R. Analytic evaluation of the cycle time on networked conflicting timed event graphs in the (max,+) algebra[J]. Discrete Event Dynamic Systems, 2016,

26(4) : 561 – 581.

[85] COTTENCEAU B, LAHAYE S, HARDOUIN L. Modeling of time-varying (max,+) systems by means of weighted timed event graphs[C] // In Proceedings of the 12th International Workshop on Discrete Event Systems. 2014 : 465 – 470.

[86] COTTENCEAU B, HARDOUIN L, BOIMOND J L. Modeling and control of weight-balanced timed event graphs in dioids[J]. IEEE Transactions on Automatic Control, 2014, 59(5) : 1219 – 1231.

[87] BERTHOMIEU B, VERNADAT F. Time Petri nets analysis with tina[C] // In Proceedings of the 3rd International Conference on Quantitative Evaluation of Systems. 2006 : 123 – 124.

[88] BONET P, LLADÓ C M, PUIJANER R, et al. PIPE v2. 5: A Petri net tool for performance modelling[C] // In Proceedings of the 23rd Latin American Conference on Informatics (CLEI 2007). 2007.

[89] JÚLVEZ J, MAHULEA C, VÁZQUEZ C-R. SimHPN: A MATLAB toolbox for simulation, analysis and design with hybrid Petri nets[J]. Nonlinear Analysis: Hybrid Systems, 2012, 6(2) : 806 – 817.

[90] SESSEGO F, GIUA A, SEATZU C. HYPENS: A Matlab tool for timed discrete, continuous and hybrid Petri nets[C] // In Proceedings of the International Conference on Applications and Theory of Petri Nets. 2008 : 419 – 428.

[91] BERTHOMIEU B, MENASCHE M. An enumerative approach for analyzing time Petri nets[C] // Proceedings IFIP. 1983.

[92] BERTHOMIEU B, DIAZ M. Modeling and verification of time dependent systems using time Petri nets[J]. IEEE Transactions on Software Engineering, 1991, 17(3) : 259 – 273.

[93] RODRIGUEZ R J, JULVEZ J, MERSEGUER J. On the performance estimation and resource optimization in process Petri nets[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2013, 43(6) : 1385 – 1398.

[94] MUNIER A. Régime asymptotique optimal d'un graphe d'évènement temporisé généralisé: application à un problème d'assemblage[M]. [S.l.] : Université Paris-Sud, Centre d'Orsay, Laboratoire de recherche en Informatique, 1992.

[95] NAKAMURA M, SILVA M. Cycle time computation in deterministically timed weighted marked graphs[C] // In Proceedings of the 7th IEEE International Conference on Emerging Technologies and Factory Automation. 1999 : 1037 – 1046.

[96] BENFEKIR A, HAMACI S, DARCHERIF M. Allocating resources of weighted T-system for adaptive behaviour[J]. International Journal of Operational Research, 2012, 14(1) : 31 – 49.

[97] TROUILLET B, BENASSER A, GENTINA J C. On the linearization of weighted T-systems[J].

International Journal of Production Research, 2008, 46(19) : 5417 – 5426.

[98] HAMACI S, BOIMOND J L, LAHAYE S, et al. On the linearizability of discrete timed event graphs with multipliers using (min,+) algebra[C] // In Proceedings of the 7th International Workshop on Discrete Event Systems. 2004 : 367 – 372.

[99] HAMACI S, BOIMOND J-L, LAHAYE S. Performance analysis of timed event graphs with multipliers using (min,+) algebra[G] // Informatics in Control, Automation and Robotics II. [S.l.] : Springer, 2007 : 185 – 190.

[100] HAMACI S, DARCHERIF A M, LABADI K. Performance evaluation of timed Petri nets in dioid algebra[M]. [S.l.] : INTECH Open Access Publisher, 2012.

[101] BENFEKIR A, HAMACI S, BOIMOND J-L, et al. Performance evaluation of nonlinear weighted T-system[J]. International Journal of Systems Science, 2013, 44(10) : 1948 – 1955.

[102] BENFEKIR A, HAMACI S, DARCHERIF A-M, et al. On the nonlinear dynamic behavior of unrelaxed timed Petri nets in idempotent semirings[C] // In Proceedings of the 18th International Conference on Methods and Models in Automation and Robotics. 2013 : 771 – 776.

[103] KAHOUADJI H, HAMACI S, LABADI K, et al. A new upper bound of cycle time in weighted marked graphs[C] // In Proceedings of the International Conference on Control, Decision and Information Technologies. 2013 : 137 – 142.

[104] TROUILLET B, BENASSER A, GENTINA J-C. Transformation of the cyclic scheduling problem of a large class of fms into the search of an optimized initial marking of a linearizable weighted t-system[C] // In Proceedings of the 6th International Workshop on Discrete Event Systems. 2002 : 83 – 90.

[105] DASDAN A. Experimental analysis of the fastest optimum cycle ratio and mean algorithms[J]. ACM Transactions on Design Automation of Electronic Systems, 2004, 9(4) : 385 – 418.

[106] KARP R M. A characterization of the minimum cycle mean in a digraph[J]. Discrete mathematics, 1978, 23(3) : 309 – 311.

[107] REITER R. Scheduling parallel computations[J]. Journal of the ACM, 1968, 15(4) : 590 – 599.

[108] YOUNG N E, TARJANT R E, ORLIN J B. Faster parametric shortest path and minimum-balance algorithms[J]. Networks, 1991, 21(2) : 205 – 221.

[109] GHAMARIAN A H, GEILEN M, STUIJK S, et al. Throughput analysis of synchronous data flow graphs[C] // In Proceedings of the 6th International Conference on Application of Concurrency to System Design. 2006 : 25 – 36.

[110] CASALE G, MI N, SMIRNI E. Bound analysis of closed queueing networks with workload burstiness[C] // ACM SIGMETRICS Performance Evaluation Review. 2008 : 13 – 24.

[111] LI R, REVELIOTIS S. Performance optimization for a class of generalized stochastic Petri nets[J].

Discrete Event Dynamic Systems, 2015, 25(3) : 387 – 417.

[112] HEE K, REIJERS H, VERBEEK E, et al. On the optimal allocation of resources in stochastic workflow nets[C] // In Proceedings of the 7th UK Performance Engineering Workshop. 2001 : 23 – 34.

[113] CHEN H, AMODEO L, CHU F, et al. Performance evaluation and optimization of supply chains modelled by batch deterministic and stochastic Petri net[J]. IEEE Transactions on Automation Science and Engineering, 2005, 2(2) : 132 – 144.

[114] ABDALLAH I B, ELMARAGHY H A, ELMEKKAWY T. Deadlock-free scheduling in flexible manufacturing systems using Petri nets[J]. International Journal of production research, 2002, 40(12) : 2733 – 2756.

[115] WU N, ZHOU M. Real-time deadlock-free scheduling for semiconductor track systems based on colored timed Petri nets[J]. OR Spectrum, 2007, 29(3) : 421 – 443.

[116] XING K, HAN L, ZHOU M, et al. Deadlock-free genetic scheduling algorithm for automated manufacturing systems based on deadlock control policy[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2012, 42(3) : 603 – 615.

[117] RODRIGUEZ-BELTRAN J, RAMFREZ-TREVINO A. Minimum initial marking in timed marked graphs[C] // In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics. 2000 : 3004 – 3008.

[118] NAKAMURA M, SILVA M. An iterative linear relaxation and tabu search approach to minimum initial marking problems of timed marked graphs[C] // In Proceedings of the European Control Conference. 1999 : 985 – 990.

[119] GIUA A, PICCALUGA A, SEATZU C. Incremental optimization of cyclic timed event graphs[C] // In Proceedings of the IEEE International Conference on Robotics and Automation. 2000 : 2211 – 2216.

[120] GIUA A, PICCALUGA A, SEATZU C. Firing rate optimization of cyclic timed event graphs by token allocations[J]. Automatica, 2002, 38(1) : 91 – 103.

[121] LAFTIT S, PROTH J M, XIE X L. Marking optimization in timed event graphs[C] // In Proceedings of the International Conference on Application and Theory of Petri Nets. 1991 : 281 – 300.

[122] GAUBERT S. An algebraic method for optimizing resources in timed event graphs[G] // Analysis and Optimization of Systes. [S.l.] : Springer, 1990 : 957 – 966.

[123] GAUBERT S. Resource optimization and (min,+) spectral theory[J]. IEEE Transactions on Automatic Control, 1995, 40(11) : 1931 – 1934.

[124] CHRÉTIENNE J C P, CARLIER J. Problème d'ordonnancement: modélisation, complexité, algorithmes[M]. [S.l.] : Masson, Paris, 1988.

[125] MILLO J V, SIMONE R D. Periodic scheduling of marked graphs using balanced binary words[J]. Theoretical Computer Science, 2012, 458 : 113 – 130.

[126] PANAYIOTOU C G, CASSANDRAS C G. Optimization of kanban-based manufacturing systems[J]. Automatica, 1999, 35(9) : 1521 – 1533.

[127] RAMAMOORTHY C V, HO G S. Performance evaluation of asynchronous concurrent systems using Petri nets[J]. IEEE Transactions on Software Engineering, 1980, 6(5) : 440 – 449.

[128] WANG H, ZENG Q. Modeling and analysis for workflow constrained by resources and nondetermined time: An approach based on Petri nets[J]. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 2008, 38(4) : 802 – 817.

[129] LOFBERG J. YALMIP: a toolbox for modeling and optimization in MATLAB[C] // In Proceedings of the IEEE International Conference on Robotics and Automation. 2004 : 284 – 289.

[130] GAREY M R, JOHNSON D. Computers and intractability: A guide to the theory of NP-completeness[M]. [S.l.] : WH Freeman and Company, New York, 1979.

Acknowledgement

First and foremost, I am greatly indebted to my two supervisors, Prof. Zhiwu Li and Prof. Alessandro Giua, for their constant encouragement, patient guidance, instructive suggestions, and financial support. Frankly speaking, without their encouragement, I can hardly start my Ph.D study at the beginning. They dedicated me a lot of time and taught me to pay attention to details in doing research and writing papers. I felt so lucky to work with them in these five years, not only for their vast knowledge and skills, but also for their respectable attitude towards research.

Except for my supervisors, I would like to show my special thanks to Mr. Guanghui Zhu, Ms. Yin Tong, and Dr. Ziyue Ma, for their thoughtful suggestions and helpful discussions in our regular seminars. I appreciate their persistence in pursing perfect works. I will always remember the time we spent together to discuss technical and programming problems.

I extend my special thanks to other professors for their help, concern, and friendship. Among them are Prof. Carla Seatzu, Prof. Kamel Barkaoui, Prof. Francesco Basile, Prof. W. M. Wonham, and Prof. Lei Feng.

I would like to express my gratitude to Prof. Isabel Demongodin, Ms. Béatrice Alcala, Dr. Maâmar el-amine Hamri, Dr. Nesrine Driouche from Aix-Marseille University, and Mr. Ning Ran from Zhejiang University. They provided me a lot of help during my stay in Marseille. Special thanks to Mr. Aiwen Lai, Mr. Changshun Wu, and Ms. Wenjing Yang. I spent a pleasure time with them in Marseille in France.

My sincere thanks also goes to my lab mates in Xidian University. It is my pleasure to conduct research with them for more than five years. Particularly among them are Dr. Ding Liu, Dr. Yifan Hou, Dr. Gaiyun Liu, Dr. Xubin Ping, Dr. Anrong Wang, Dr. Meng Qin, Dr. Jiafeng Zhang, Dr. Xiaoliang Chen, Dr. Xi Wang, Dr. Zhongyuan Jiang, Mr. Changming Shao, Mr. Jingyang Fan, Mr. Chao Gu, Mr. Chao Wang, Mr. Pei Li, Mr. Hang Xu, Mr. Kaijie Xu, Mr. Kuangze Wang, and Mr. Xuya Cong. I want to thank, especially, Dr. Miao Liu, Dr. Xiuyan Zhang, Mr. Xiang Gao, Mr. Wei Chen, Mr. Deguang Wang, Ms. Chan Gu, Ms. Lan Yang, Mr. Liang Li, Mr. Dongdong Guan, Mr. Haibo Wang, Mr. Chao Sun, Ms. Siyan Chen, Mr. Bing Hui, Mr. Chengzong Li, and Mr. Wenlong He.

I would like to thank all my friends in Xidian University. Particularly among them are Mr. Yan Ma, Mr. Jiyong Meng, Mr. Xiaodong Shang, Mr. Wentao Zhao, Mr. Naigang Hu, Mr. Xing Quan, and Mr. Gangshan Jing. They are always full of enthusiasm.

I am truly grateful to my beloved grandparents, parents, and my sister, for their constant encouragement, love, and caring through my life. My dear grandmother always comforts me when I feel frustrated. My love for her means more than I can ever express. Finally, I would like to express appreciation to my beloved girl friend Ms. Huiru Yang. She always supported me during this period and gave me the right strength to live in happiness for last six years. I will spend all of my lifetime to love and protect her.

Xidian University, China *Zhou He*

March 22, 2017

Biography

1. Basics

Zhou He (male) was born in Ankang, Shaanxi province, China, in January 1990. He received Bachelor Degree in College of Mechanical and Electrical Engineering from Shaanxi University of Science and Technology, Xi'an, China, in 2012. From September 2012 to July 2013 he has been a master candidate of School of Electro-Mechanical Engineering of Xidian University and is majored in Electro-Mechanical Engineering, co-tutored by Prof. Dr. Zhiwu Li and Prof. Dr. Alessandro Giua. Since September 2013 he has been a Ph.D candidate of School of Electro-Mechanical Engineering of Xidian University and is majored in Electro-Mechanical Engineering, co-tutored by Prof. Dr. Zhiwu Li and Prof. Dr. Alessandro Giua.

2. Education Background

2008.09 $\sim$ 2012.07, Shaanxi University of Science and Technology, B.A. in Mechanical Engineering

2012.09 $\sim$ 2013.07, Xidian University, M.A. student in Electro-Mechanical Engineering (Master-doctor postgraduate student program)

2013.09 $\sim$ present, Xidian University, Ph.D student in Electro-Mechanical Engineering (Master-doctor postgraduate student program)

2014.09 $\sim$ present, Aix-Marseille University, Ph.D student in Automatique

3. Academic Publications

3.1 Journal Publications

[1] **Z. He**, Z. W. Li, A. Giua. "Optimization of Deterministic Timed Weighted Marked Graphs", *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 1084-1095, 2017. (SCI Journal, JCR Q1)

[2] **Z. He**, Z. W. Li, A. Giua. "Cycle time optimization of deterministic timed weighted marked graphs by transformation", *IEEE Transactions on Control Systems Technology*, vol. 25, no. 4, pp. 1318-1330, 2017. (SCI Journal, JCR Q1)

3.2 Conference Publications

[1] **Z. He**, Z. W. Li, A. Giua. "Marking Optimization of Deterministic Timed Weighted Marked Graphs", In *Proceedings of the 10th IEEE International Conference on Automation Science and Engineering*, 2014: 413-418. (EI: 20153501210506)

[2] **Z. He**, Z. W. Li, A. Giua. "Cycle Time Optimization of Deterministic Timed Weighted Marked Graphs", In *Proceedings of the 11th IEEE International Conference on Automation Science and Engineering* (CASE'15), 2015: 274-279. (EI: 20160101765651)

[3] **Z. He**, Z. W. Li, I. Demongodin, A. Giua. "Marking optimization of deterministic timed weighted marked graphs under infinite server semantics", In *Proceedings of the 3rd International Conference on Control, Decision and Information Technologies*, (CoDIT'16), 2016: 1-6. (EI: 20164703033928)

[4] **Z. He**, Z. W. Li, A. Giua. "Stationary behavior of manufacturing systems modeled by timed weighted marked graphs", In *Proceedings of the IEEE Region 10 Conference* (TENCON'16), 2016: 3374-3377 (EI: 20171203467721).

[5] **Z. He**, Z. W. Li, A. Giua. "Cycle time optimization for deterministic timed weighted marked graphs under infinite server semantics", In *Proceedings of the 55th IEEE Conference on Decision and Control* (CDC'16), 2016: 3942-3947 (EI: 20170503304865).

3.3 Participation in Academic Programs

[1] General project of National Natural Science Foundation of China under Grant No. 61374068 entitled "Optimal liveness-enforcing Petri net supervisors for automated manufacturing systems based on structural analysis", 2014.1-2017.12, Ongoing.

[2] General project of National Natural Science Foundation of China under Grant No. 61472295 entitled "Supervision and Reconfiguration in Discrete Event Systems", 2015.1-2018.12, Ongoing.