



HAL
open science

On intrinsically live structure and deadlock control of generalized Petri nets modeling flexible manufacturing systems

Ding Liu

► **To cite this version:**

Ding Liu. On intrinsically live structure and deadlock control of generalized Petri nets modeling flexible manufacturing systems. Modeling and Simulation. Conservatoire national des arts et metiers - CNAM, 2015. English. NNT : 2015CNAM1004 . tel-01296468

HAL Id: tel-01296468

<https://theses.hal.science/tel-01296468>

Submitted on 31 Mar 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

École Doctorale Informatique, Télécommunications et Electronique de Paris

Centre d'Etudes et De Recherche en Informatique du CNAM

THÈSE DE DOCTORAT

présentée par : Ding LIU

soutenue le : 8 Juillet 2015

pour obtenir le grade de : Docteur du Conservatoire National des Arts et Métiers

Discipline / Spécialité : Informatique

INTRINSICALLY LIVE STRUCTURES AND DEADLOCK CONTROL IN GENERALIZED PETRI NETS

THÈSE DIRIGÉE PAR

M. BARKAOUI Kamel

Professeur des universités, Cedric, Le CNAM, France

M. LI Zhiwu

*Professeur des universités, SCA Group, Xidian Université,
Chine*

RAPPORTEURS

M. ALLA Hassane

*Professeur des universités, GIPSA Lab., Université Joseph
Fourrier, France*

M. COUVREUR Jean-Michel

*Professeur des universités, LIFO, Université d'Orléans,
France*

EXAMINATEURS

Mme. CHU Feng

*Professeur des universités, IBISC, Université d'Evry Val
d'Essone, France*

M. LOISEAU Jean-Jacques

*Directeur de recherches au CNRS, IRCCyN, Université de
Nantes, France*

Mme. BÉRARD Béatrice

Professeur des universités, LIP6, UPMC, France

Remerciements

First, thanks to the following program and organizations. Due to the “Programme Cai Yuanpei” initiated by both Campus France and Chinese Scholarship Council, I can, fortunately, be a part of the joint training program and the cooperation between le Cnam, France and Xidian University, China. In Paris, le Cnam hosted me here for nearly two and half years, which is an amazing and unforgettable experience in my life. In Xi’an, Xidian University provided me a teaching position so that I can stay in the same place where I was since when I left my hometown less than 18 years old. And, I also want to thank the National Natural Science Foundation of China for providing the young researcher funds, which is the first and an important step for me to continue to walk on the pathway of academic research.

Among all persons, I must thank my family first. In memory of my grandmother, may her rest in peace. She told me to be strong and hoped me to be a professor in a university. Now, I took the first step. I am deeply grateful to my grandfather, parents, aunts, uncles, and my little cousins, for their understanding and long-time support, especially, my mother - she did so many for me. I hope all of them healthy and happy everyday.

I would like to express my sincere appreciation to two advisors of the thesis, Prof. Kamel Barkaoui and Prof. Zhiwu Li. I learned a lot from their guidance and encouragement. I first met Prof. Barkaoui in Xi’an in 2008 for the 29th International Conference on Application and Theory of Petri Nets and Other Models of Concurrency, of which I was a volunteer. During my sojourn in Paris, he helped me a lot that I cannot list here due to the limited space. “Be zen and be quiet.” He told me when I was very depressed and upset about the heavy work and limited time when working on the thesis. I also learned a lot when I assisted him in organizing and holding some international workshops and conferences, like FM2012, BPM-PALS2013, and PN2014. Since 2003, I have been with Systems Control and Automation Group led by Prof. Li in Xi’an for almost twelve years. It is him who introduced me to the exciting research field and advanced my professional career. He sets a good example for me by showing his meticulous attention to details, hard working spirit, and tenacity in goal pursuit.

I extend my gratitude to other professors for their help, concern, and friendship. From 2007 to 2010, Prof. Mengchu Zhou of New Jersey Institute of Technology was one of the Chang Jiang Scholars of Xidian University. I got a well guidance from him and even today I still remember some valuable suggestions in his first speech in Xidian University. He opened my mind and gave me many comments and suggestions on my every paper. He is a mentor of mine. Prof. Yisheng Huang of National Ilan University, like a big brother, gave me a lot of advices on research and life during our many times meets in Xi’an, Hangzhou, and Paris. From 2012, Prof. Alessandro Giua works as a guest professor in our research group in Xi’an. He gave genuine helps and some valuable suggestions on my work.

I am truly grateful to all the fellows in the laboratories at le Cnam and Xidian University. Particularly among them are Dr. Borhan Marzougui, Dr. Réda Benosman, Dr. Anrong Wang, Dr. Mi Zhao, Dr. Yifan Hou, Dr. Meng Qin, Dr. Qi Wang, Ms. Jiafeng Zhang, Ms. Yin Tong, Mr. Zhou He, Ms. Lan Yang, and Mr. Deguang Wang. I want to thank, especially, Dr. Yufeng Chen and Dr. Gaiyun Liu. We stayed in Paris and finished the same joint training program together. They cooked very well and I enjoyed every meal.

Résumé

Nos travaux portent sur l'analyse des systèmes de production automatisée à l'aide de réseaux de Petri. Le problème posé est de savoir si un système peut se bloquer complètement ou partiellement et si besoin de calculer un contrôleur garantissant son bon fonctionnement. Les systèmes de production se modélisent naturellement à l'aide d'une sous-classe des réseaux de Petri, les S^3PR . Ce modèle a été très largement étudié par le passé conduisant à des méthodes basées uniquement sur la structure du modèle. Dans ce travail, nous généralisons ces travaux aux modèles des WS^3PR , une extension des S^3PR où la réalisation d'une activité nécessite non par une ressource mais plusieurs ressources d'un même type et pour lesquels nous proposons des techniques originales combinant des éléments de théorie des graphes et de théorie des nombres, améliorant même les méthodes du passé sur le modèle simple des S^3PR .

On présente une caractérisation fine de la vivacité d'un tel modèle basée la notion d'attente circulaire. Une attente circulaire peut être vue comme une composante connexe du sous graphe réduit aux transitions et aux places ressources du modèle. Puis nous démontrons que la non vivacité d'un WS^3PR est équivalente à l'existence d'« un blocage circulaire dans une attente circulaire ». Ce résultat généralise finement la caractérisation de la vivacité d'un S^3PR . Après avoir introduit la notion de « circuits du graphe de ressources » (WSDC), on construit une méthode de contrôle de ces verrous garantissant la vivacité du modèle d'autant plus efficace qu'une méthode de décomposition du réseau est proposée. Enfin, une traduction de traduit la condition de vivacité des WS^3PR sous la forme d'un programme linéaire en nombres entiers est établie et des expérimentations ont démontré l'intérêt de la méthode pour contrôle de systèmes l'allocation des ressources.

Mots clés : Systèmes de production flexible, Réseaux de Petri généralisés, Analyse structurelle, Vivacité structurelle, Contrôle d'interblocage

Abstract

As an indispensable component of contemporary advanced manufacturing systems, flexible manufacturing systems (FMSs) possess flexibility and agility that traditional mass production systems lack. An FMS usually consists of picking and placing robots, machining centers, logistics systems, and advanced control systems. Some of them can be recognized as the FMS' shared resources, which result in its flexibility but may also lead to its deadlocks. As a classic problem in resource allocation systems, deadlocks may arise in a fully automated FMS and bring about a series of disturbing issues, from degraded and deteriorated system productivity and performance to low utilization of some critical and expensive resources and even long system downtime. Therefore, the analysis of and solution to deadlock problems are imperative for both a theoretical investigation and practical application of FMSs. Deadlock-freedom means that concurrent production processes in an FMS will never stagnate. Furthermore, liveness, another significant behavioral property, means that every production process can always be finished in finite time. Liveness implies deadlock-freedom, but not vice versa. The liveness-enforcement is a higher requirement than deadlock-freedom.

Completely without reachability graph, the work is based on the logical reasoning of model structures. From the perspective of the behavioral logic, the thesis focuses on the intrinsically live structures and deadlock control of generalized Petri nets modeling flexible manufacturing systems. Being different from the prevailing structure/siphon-based methods, a concept of intrinsically live structures becomes the starting point to design, analyze, and optimize a series of novel deadlock control and liveness-enforcing methods in the work. The characteristics and essence of intrinsically live structures are identified and derived from subclasses of generalized Petri nets modeling FMSs with complex resource usage styles. In addition, the numerical relationship between initial

ABSTRACT

markings and weights of connecting arcs is investigated and used to design restrictions that can ensure the intrinsic liveness of global or local structures. With the structural theory, graph theory, and number theory, the work in the thesis achieves the goals of deadlock control and liveness-enforcement. The proposed methods are superior over the traditional siphon-based ones with a lower computational complexity (or a higher computational efficiency), a lower structural complexity, and a better behavioral permissiveness of the controlled system.

Keywords : flexible manufacturing system, Petri net, generalized Petri net, live structure, deadlock control

Table des matières

Remerciements	3
Résumé	5
Abstract	7
Liste des tableaux	13
Table de figures	18
Introduction	19
0.1 Motivation	19
0.2 Flexible manufacturing systems, deadlock control, and Petri nets	22
0.3 Thesis Organization	27
0.3.1 Publications	29
1 Preliminaries of Petri Nets	33
2 Systems of Simple Sequential Processes with Weighted Resource Allocation	39
2.1 WS^3PR	39
2.2 Circular wait and circular blocking in WS^3PR	44
3 Intrinsically Live Structure	49

TABLE DES MATIÈRES

3.1	Introduction	49
3.2	Intrinsically live structure	50
3.3	Algorithm and examples	59
3.4	Discussion	68
3.5	Summary	72
4	Hybrid Liveness-enforcing Method for WS³PR	73
4.1	Introduction	73
4.2	Liveness-enforcement in WS ³ PR	74
4.2.1	Liveness-enforcement based on elementary siphons	74
4.2.2	Liveness-enforcement based on intrinsically live structures	77
4.3	Hybrid liveness-enforcing policy	84
4.4	Summary	91
5	Parameterized Liveness and Ratio-enforcing Supervisor for WS³PR	93
5.1	Introduction	93
5.2	Liveness and ratio-enforcement in WS ³ PR	95
5.2.1	Resource usage ratio	95
5.2.2	Ratio analysis of an intrinsically live structure	98
5.2.3	Synthesis of liveness and ratio-enforcing supervisors	103
5.3	Discussion	108
5.4	Summary	110
6	Liveness-enforcement for WS³PR Using a Divide-and-Conquer Paradigm	113
6.1	Introduction	113
6.2	Dived-and-conquer strategy	115
6.2.1	Primary separation of toparchies	117

TABLE DES MATIÈRES

6.2.2	Advanced separation of toparchies	120
6.2.3	Toparch and controlled WS ³ PR synthesis	125
6.3	Discussion	128
6.4	Summary	131
7	Iterative Liveness-enforcement for WS³PR	133
7.1	Introduction	133
7.2	Liveness and ratio-enforcing supervisor	134
7.3	MIP and LRS	135
7.4	Summary	139
8	On Intrinsically Live Structure of GS³PR Modeling FMS	141
8.1	Introduction	141
8.2	GS ³ PR	142
8.3	ILS in GS ³ PR	146
8.4	Summary	153
9	ILS in Process and Resource-Oriented Petri Nets	155
9.1	SRS – a class of ROPN	155
9.2	ILS in SRS	158
9.3	Summary	161
10	Conclusions and Future Research	163
10.1	Contributions	163
10.2	Future Work	166
	Glossaire	185

TABLE DES MATIÈRES

Liste des tableaux

3.1	WSDCs and their respective competition paths that satisfy Restriction 3.1 in the WS ³ PR of Fig. 3.10	69
3.2	The minimal $M_0(r_i)$ for resource places used by competition paths in Tabel 3.1	69
5.1	Full details about RU-ratios of WS ³ PR net in Fig. 5.2(a)	98
6.1	Different control effects of the LRS monitor v_3 with different control parameters for WS ³ PR shown in Fig. 6.4(a).	127
7.1	Details of the iterative control of the WS ³ PR net shown in Fig. 7.3.	139
8.1	A Comparison between the control effects of the elementary siphons-based method (ES) in [120] and the intrinsically live structures-based one (ILS) in this chapter for the GS ³ PR net model shown in Fig. 8.4(a).	152

LISTE DES TABLEAUX

Table des figures

1	A Chinese watercolor painting showing two oxen falling into a deadlock on a single plank bridge (Drawn by Ms. Qing Zhao)	19
2	Two heavy trucks are stuck in a narrow corner of a road. (Photo by Ms. Jianying Shi in Gansu province, China)	20
3	(a) Two oxen on the single plank bridge, (b) two jobs requiring the same amount of the whole capacity of a resource, (c) one job requiring $2/3$ of the whole capacity and the other requiring $1/2$, and (d) one job requiring $2/3$ and the other requiring $1/3$.	21
4	(a) Only one kind of jobs for a resource, (b) two kinds of jobs for the resource, and (c) three kinds of jobs for the resource.	22
5	A histogram showing the numbers of items searched by Google scholar with the search expression “allintitle : petri net OR nets -dishes -dish -plate -plates ” for the past 50 years. Final data obtained on July 15, 2015.	24
6	The framework of the thesis.	27
1.1	(a) An ordinary net, (b) a generalized net, and (c) a weighted digraph of resource places, G_R	34
1.2	(a) A generalized net and (b) one of its subnets.	36
2.1	(a) An S^2PR (S^3PR) net model and (b) a WS^2PR (WS^3PR) net model.	40
2.2	(a) A generalized net and (b) one of its subnets.	42
2.3	An S^3PR model for the problem of Dining Philosophers.	44

TABLE DES FIGURES

3.1	(a) A marked WS ³ PR net model and (b) a corresponding weighted digraph of resource places.	51
3.2	A general WSDC.	52
3.3	(a) A competition path and (b) a subnet containing the competition path and its upstream and downstream activity places.	53
3.4	Multiple competition paths with a same resource place r	54
3.5	A WS ³ PR net with two shared resource places.	55
3.6	A WS ³ PR net model and two competition paths with the resource place p_{10}	56
3.7	(a) A WS ³ PR net model N with 5 resource places and (b) G_R derived from N	60
3.8	There exists a proper initial marking of p_{17} to satisfy Restriction 3.1.	63
3.9	A WS ³ PR with six resource places.	67
3.10	A WS ³ PR model with 9 WSDCs in its G_R	68
3.11	A WS ³ PR modeling the problem of extended Dining Philosophers.	71
4.1	(a) A WS ³ PR with five resource places, (b) four monitors designed according to elementary siphons, and (c) the corresponding weighted digraph of resource places.	76
4.2	(a) An ordinary net, (b) a generalized net, and (c) a weighted digraph of resource places, G_R	78
4.3	A more complex WS ³ PR net model with two highly shared resource places.	82
4.4	A WS ³ PR net model with 6 resource places.	85
5.1	A WS ³ PR net with two shared resource places.	95
5.2	(a) A generalized net and (b) one of its subnets.	96
5.3	(a) A competition path and (b) a subnet containing the competition path and its upstream and downstream activity places.	99
5.4	Multiple competition paths with a same resource place r	99
5.5	(a) A WS ³ PR net with initial marking $M_0 = 50p_1 + 50p_4 + 50p_7 + 5p_{10} + 8p_{11}$ and (b) two WSDCs, each of which contains a competition path satisfying Restriction 3.1.	100

TABLE DES FIGURES

5.6	(a) A control path and (b) the control path and corresponding paired arcs imposed on a subnet containing a competition path.	103
5.7	A general WSDC with an LRS monitor v for resource place r_2	103
5.8	(a) An S^3PR net model to be controlled and (b) an LRS synthesized for the net.	105
6.1	(a) A WS^3PR net model to be controlled , (b) a WSDC , (c) a subnet containing a strict minimal siphon, (d) an LRS monitor, and (e) a traditional siphon-based monitor.	116
6.2	A WS^3PR net model.	118
6.3	(a) A generalized Petri net (N_1, M_{01}) , (b) another generalized Petri net (N_2, M_{02}) , (c) a composed net (N, M_0) , and (d) a subnet of (N, M_0)	121
6.4	(a) A WS^3PR net with only one toparchy and (b) its weighted digraph of resource places.	122
6.5	(a) A subnet after removing nodes and arcs of WSDC $t_3r_3t_4r_2t_3$ that do not belong to any other WSDC, (b) after removing nodes and arcs of WSDC $t_9r_7t_{15}r_6t_9$ that do not belong to any others, and (c) the final remaining subnet to be controlled after removing nodes and arcs of WSDC $t_{10}r_6t_{14}r_5t_{10}$ that do not belong to any others...	124
6.6	(a) An LRS monitor v_1 for r_5 shown in Fig. 6.2, (b) an LRS monitor v_2 for r_1 shown in Fig. 6.5(c), and (c) an alternative LRS monitor v_3 for r_4 shown in Fig. 6.5(c).	126
6.7	A schematic diagram of the D&C liveness-enforcing method for WS^3PR	129
7.1	A competition path and its imposed control one.	134
7.2	A schematic diagram of the iterative method.	136
7.3	A WS^3PR net model.	137
7.4	Three LRS monitors.	138
8.1	Classes of Petri nets.	143
8.2	(a) A Petri net model and (b) one of its subnets.	143

TABLE DE FIGURES

8.3	(a) An arc pair formed by (r, t_1) and (t_{k+1}, r) with $k \geq 1$ and (b) two examples of resource usage styles not permitted in GS^2PR	145
8.4	(a) A marked GS^3PR net model and (b) its weighted digraph of wait relations. . .	147
8.5	Two WSDCs of the GS^3PR net shown in Fig. 8.4(a) : (a) a WSDC including activity places, and (b) another WSDC not including any activity place.	148
8.6	(a) A competition path $t_i r t_h$ and (b) its sets of upstream and downstream activity places P_{up}^r and P_{down}^r	149
9.1	(a) An RS N_1 processing A-type of workpieces and (b) another RS N_2 processing B-type of workpieces.	156
9.2	An SRS obtained by merging N_1 and N_2 in Fig. 9.1.	158
9.3	(a) A weighted digraph of resource places of the SRS in Fig. 9.2 and (b) two WSDCs derived from the SRS.	159
9.4	A CPN-Tools [7] model of the SRS in Fig. 9.2.	160
10.1	The main work of the thesis.	164
10.2	A wrong result given by Lingo.	168

Introduction

0.1 Motivation

In an informal way, an analogy is used to explain the beginning of the whole story and the motivation of the research work in the thesis. As shown in Fig. 1, two oxen appear at both ends of a single plank bridge in the Chinese watercolor painting. Both of them want to pass through the bridge, however, the plank is not wide enough for them to go across the bridge at the same time. In addition, in our Chinese culture, we think that oxen are very stubborn animals. Each of them stands still and refuses to make way for the other. They, unfortunately, fall into an impasse or a so-called deadlock.



FIG. 1 – A Chinese watercolor painting showing two oxen falling into a deadlock on a single plank bridge (Drawn by Ms. Qing Zhao)

The same situation of the two oxen scenario happens in the real world. For example, as shown in Fig. 2, two heavy trucks are stuck in a narrow corner of a road. The road is merely wide for only

0.1. MOTIVATION

one of them going through, however, both of them want to go across in the opposite directions at the same time. Obviously, from the viewpoint of system theory, the so-called resource here in such a system, i.e., the road, is limited (not wide enough) and not allocated properly (a truck holds one end and requires the other one of this part of road, which is held by its competitor). Please note that, the two trucks scenario also implies that the lack of information/communication can be one of the reasons of this impasse (or deadlock). If the drivers can see each other before they are going to go across the narrow corner, the trouble may be avoided. To be clear in advance, the research in the thesis does not explicitly discuss the role of information/communication played in deadlocks, however, actually, the lack of information/communication is abstracted into a part of pure cause and effect relations. Similarly, the time factor is also implicit and hidden behind the cause and effect relations in formal models in the work of the thesis.



FIG. 2 – Two heavy trucks are stuck in a narrow corner of a road. (Photo by Ms. Jianying Shi in Gansu province, China)

Of course, the drivers of the heavy trucks will find a way to resolve the impasse. And the two

0.1. MOTIVATION

oxen may also solve their problem by following their animal instinct. But, what can we learn from the two scenarios? What if the same problem happens in an automated manufacturing system, which runs by itself? Is it smart enough to avoid the impasse or can it resolve the impasse after its appearance?

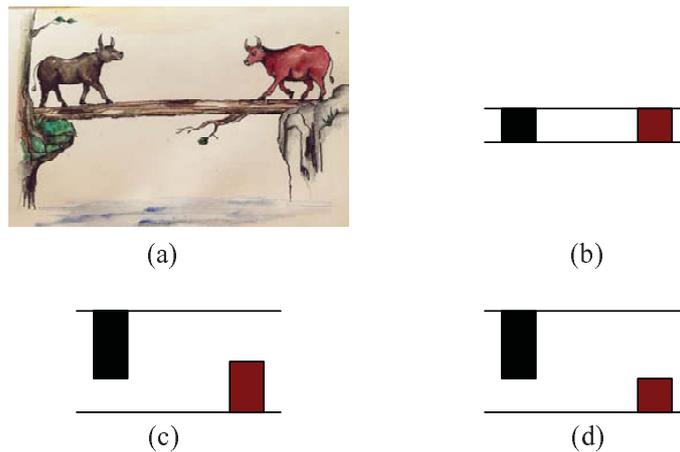


FIG. 3 – (a) Two oxen on the single plank bridge, (b) two jobs requiring the same amount of the whole capacity of a resource, (c) one job requiring $2/3$ of the whole capacity and the other requiring $1/2$, and (d) one job requiring $2/3$ and the other requiring $1/3$.

Let us go back to the two oxen scenario as shown in Fig. 3(a) for a while, and a characteristic can be found. Because it is a single plank bridge, each ox needs the entire width of the plank to go across. If we abstract it into a simple formal model shown in Fig. 3(b), we can find that two jobs in the system require the same amount, unfortunately the full, of the whole capacity of the system resource. Further, what if two jobs in the system require different amounts of units of the systems? One job requires $2/3$ of the whole capacity of the resource, whereas another job requires $1/2$ in Fig. 3(c). If the two jobs are oxen or trucks, they will face the same problem, however, they may go smooth if that one requires $2/3$ and the other requires $1/3$ as shown in Fig. 3(d).

Now, let us imagine more complex situations. Let us assume there is a resource whose capacity is $4 \times 4 = 16$. The smallest unit of the resource is $1 \times 1 = 1$ (referred as $J1$). As shown in Fig. 4(a), if each job in a system requires the smallest unit of the resource, we can see that the full capacity of the resource can be completely utilized. Please note that if the jobs requires the same amount of resource units, we call them the same kind of jobs. In the following chapters, if different work pieces to be machined according to different specifications require the same amount of resource

units, from the viewpoint of resource usage ratio, we can call them the same kind of jobs.

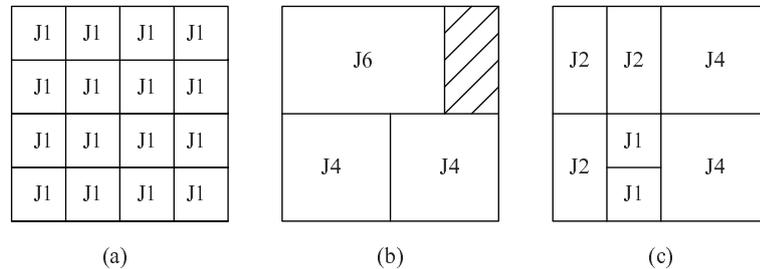


FIG. 4 – (a) Only one kind of jobs for a resource, (b) two kinds of jobs for the resource, and (c) three kinds of jobs for the resource.

It is easy to imagine that a bottle can be filled with sand and no space left in the bottle. what if you fill the bottle with some small stones ? There may be some space left but you cannot put any stone in the bottle, however, sand can be used again to put in the bottle to fill the empty space. As shown in Fig. 4(b), two kinds of jobs, $J4$ ($2 \times 2 = 4$) and $J6$ ($2 \times 3 = 6$) are using the same resource whose capacity is 16. Obviously, the shadow area is the empty space left. Further, as shown in Fig. 4(c), if three kinds of jobs, $J1$, $J4$, and $J2$ ($1 \times 2 = 2$) are simultaneously using the resource, no space left. It means that an impasse may not happen and the resource can be fully utilized.

When we study the deadlock control problem in automated manufacturing systems modeled by Petri nets, the situation shown in Fig. 4(a) can be thought as the ordinary Petri net models, whose arc weights are all equal to one, whereas the situations shown in Figs 4(b) and (c) can be thought as the generalized Petri net models, whose arc weights can be equal or greater than one. A great deal of work has been developed for the deadlock control in ordinary Petri nets, however, relatively few work focuses on that in generalized Petri nets. The work in the thesis is to investigate the deadlock control and liveness enforcing problem in generalized Petri nets with structure analysis and without reachability graph.

0.2 Flexible manufacturing systems, deadlock control, and Petri nets

In this new millennium, the global economy becomes increasingly inseparable. As a paramount foundation of our society's economy system, manufacturing encounters more and more intense global market competitions. It is no longer only directed toward the mass production of products,

especially in the high value-added and technology-intensive fields. With the rise of customizability, a high-mix-low-volume production mode [68] becomes an irresistible trend and is used by more and more companies to adapt themselves to the social development. It is the very reason that flexible manufacturing systems (FMSs) are built to realize a rapid and mixed production.

As an indispensable component of contemporary computer integrated manufacturing systems (CIMs), FMSs are constructed with flexibility and agility that traditional manufacturing systems lack to produce mixed types of products at the same time in a rapid way. They can thus react swiftly to the changes of production requirements, whether planned or unplanned, to satisfy different market segments with reasonably priced and customized products.

An FMS usually consists of picking and placing robots, machining centers with multiple capabilities and functions, logistic systems, and advanced control systems. Some of the components can be viewed as resources of the system, such as robots, machining centers, and logistic systems. The resources can be dynamically configured and allocated before and during the system operation according to changeable product specifications. Hence, in this sense, an FMS belongs to resource allocation systems (RASs) [82] [94]. The very origin of the flexibility is the sharing of the limited resources. By sharing the resources, multiple different raw workpieces are concurrently processed in the FMS to meet today's manufacturing need for high-mix-low-volume products. Due to the competition for limited resources among concurrently-executed processes, deadlocks arising in the FMS bring about a series of disturbing issues, from degraded and deteriorated system productivity and performance to low utilization of some critical and expensive resources and even long system downtime. Therefore, as a classic problem, deadlocks must be resolved for FMSs.

Besides automata [81] [104] and graph theory [13] [19] [18], Petri nets [79] [128] [127] are an effective formalism and used extensively for modeling and analyzing deadlock problems in FMSs. The relationship between shared resources and deadlocks is fully investigated and reported by many researchers [15]. The essential reason of deadlocks in an FMS is identified as improper allocation of limited resources, which leads to the occurrence of circular waits. Therefore, deciding how to properly dispatch the shared resources to meet certain control constraints becomes the starting point of most deadlock prevention policies. Corresponding to deadlocks, liveness in Petri

0.2. FLEXIBLE MANUFACTURING SYSTEMS, DEADLOCK CONTROL, AND PETRI NETS

nets, which implies deadlock-freedom but not vice versa, is a significant behavioral property in the research realm of FMSs. It means that every production process can always be finished and it is always possible to introduce new raw parts in the system [16]. The relationship between shared resources and deadlock/liveness problems is well investigated and reported in literature.

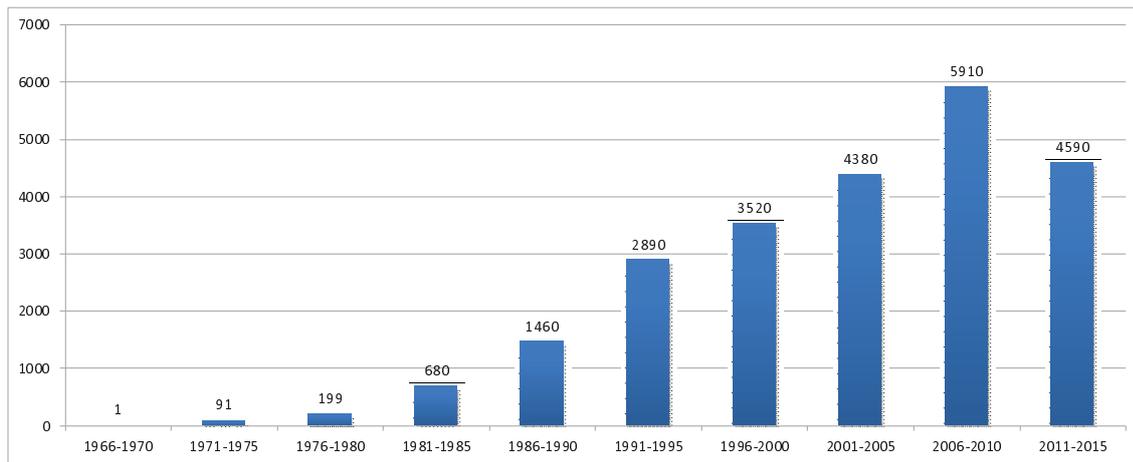


FIG. 5 – A histogram showing the numbers of items searched by Google scholar with the search expression “allintitle : petri net OR nets -dishes -dish -plate -plates ” for the past 50 years. Final data obtained on July 15, 2015.

In 1962, Carl A. Petri published his Ph.D thesis, *Kommunikation mit Automaten*, University of Bonn. It is the foundation stone of the field. From then on, the theory and applications of Petri nets grow rapidly during the past half century. The data in a histogram, shown in Fig. 5, is a rough proof of the development of research with Petri nets. By using Google Scholar, the numbers of items of every five years searched with the search expression “allintitle : petri net OR nets -dishes -dish -plate -plates” are given out. Please note that the expression “-dishes -dish -plate -plates” is used to exclude the medical research related to “Petri dishes” and similar notions. It shows the Petri net research is still increasingly growing up.

From different angles, the modeling framework of FMSs with Petri nets can be assorted into two types, resource-oriented Petri nets (ROPNs) [106]-[112] and process-oriented Petri nets (POPNS) [2] [9] [16] [37] [78] [127]-[130]. Compared with each other, an ROPN is more compact, while a POPN is more intuitive. Within the latter, the systems of simple sequential processes with resources (S^3PR) [16] [17] are a class of ordinary Petri nets widely used to model and analyze

the deadlock and liveness problems in FMSs. From the perspective of resource allocation, FMSs modeled with Petri nets can be classified into three categories according to the number and type of resource units required when a workpiece is processed at an operation step : (1) only one unit of a single type of resources, (2) one or multiple units of a single type of resources, and (3) one or multiple units of one or multiple types of resources. Since the weights of the arcs connecting activity and resource places are equal to one, an FMS modeled by S^3PR falls into the first category. That is to say, in an S^3PR , one token of only one resource place is used to complete a process of a workpiece represented by a token in an activity place. Subsequently, multiple extensions to S^3PR , such as a system of simple sequential processes with general resource requirements (S^3PGR^2) [9] [78], a weighted system of simple sequential processes with several resources (WS^3PSR) [91], a system of simple sequential processes with weighted resources allocation (WS^3PR) [64] [122], a system of sequential systems with shared resources (S^4R) [93], and G-systems [58] [130], are reported in literature, which fall into the second and third categories, i.e., they are all generalized Petri nets.

Under the tremendous influence of supervisory control theory (SCT) [81], a variety of external supervisors including monitors and connecting arcs are usually designed to impose control on the plant net model of an FMS for both ordinary and generalized Petri nets to achieve a liveness-enforcing system. Both reachability graphs [1] [11] [12] and structural analysis [2] [3] [4] [13] [16] are the most frequently employed methodologies to synthesize supervisors for plant models.

The theory of region [23] [97] [98] based on a reachability graph is a reliable, effective, and accurate method to analyze and verify a plant model. Every single state can be probed by inspecting a complete state space to yield a well-designed supervisor ensuring liveness and even maximal behavioral permissiveness. However, the cost to achieve the above goal is computationally prohibitive. The enumeration of all states is computationally expensive or infeasible in practice when dealing with a sizable plant model. The number of states of a Petri net model grows exponentially with respect to its size and initial marking. Therefore, structural theory [52] [61] [64] [65] [66] is widely used to circumvent the state explosion problem. In order to thoroughly investigate and fully utilize the structure information of a Petri net model, a special structural object, siphons, is frequently and extensively used to reveal the relationship between resources and liveness. Siphon-based characterization of deadlocks and liveness becomes a mainstream trend of methodologies

that deal with the deadlock control and liveness-enforcement issues.

Note that ordinary and generalized Petri nets have the same modeling power, but the latter can provide a much more compact and clearer expression than the former. However, due to existence of weighted arcs in generalized Petri nets, the traditional siphon-monitor-based deadlock control and liveness-enforcement developed for ordinary ones encounter many difficulties. The reason is that a siphon is defined as a set of places and does not carry the weight information of any arc. A lot of deadlock control and liveness-enforcing methods originally proposed for ordinary Petri nets cannot be directly used to generalized ones or result in poor control effects when applicable.

Weighted systems of simple sequential processes with resources (WS^3PR) [64] are a generalized extension to S^3PR . Compared with S^3PR , it can concisely model the segmentation of resource allocation. For example, a buffer has five slots, and an operation needs three of them at a time. It is convenient and clear to describe this specification by adopting a WS^3PR rather than S^3PR that needs more places, transitions, and arcs. Siphons are a special structural object that is extensively adopted to design various deadlock control methods in the literature. The insufficiently marked states of siphons are closely linked with the deadlock and non-liveness in generalized Petri net plant models. In WS^3PR as a kind of process-oriented Petri net models, an insufficiently marked siphon is composed of two types of places, i.e., empty activity places, and resource places whose marking is not enough to enable any one of their output transitions. Since the existence of non-unity arc-weights and lack of weight information in the concept of siphons, siphon-based deadlock control and liveness-enforcing methods are far away from being mature when dealing with generalized Petri nets. Aiming to improve them, the structure of circular waits (CWs) and the state of circular blockings (CBs) are both employed to investigate deadlock prevention and liveness enforcement problems, and on the ground of CWs and CBs, the concept of weighted simple directed circuits (WSDCs) is proposed in the thesis to describe the structure of a circular wait with weights, which is shown to be tied to the occurrence of deadlocks instead of the traditional structural object.

0.3 Thesis Organization

From the perspective of the behavioral logic, the work in the thesis focuses on the intrinsically live structures and deadlock control of generalized Petri nets modeling flexible manufacturing systems. Being different from the existing siphon-based methods, the concept of intrinsically live structures (ILSs) based on WSDC becomes the starting point to analyze, control, optimize, simulate, and validate novel deadlock control and liveness-enforcing methods. The characteristics of ILSs are identified and derived from subclasses of generalized Petri nets modeling FMS with complex resource usage styles. Additionally, the numerical relationship between initial markings and weights of connecting arcs is investigated and used to design a restriction that ensures the intrinsic liveness of local structures. With the structural theory, graph theory, and number theory, the work achieves the goals of deadlock control and liveness-enforcement superior over the traditional siphon-based ones with a lower computational complexity (or a higher computational efficiency), a lower structural complexity, and a better behavioral permissiveness of the controlled system. The main work of the thesis is shown in Fig. 6 and the organization is listed as follows. Please note that the analysis of intrinsically live structures in WS^3PR is the very beginning of all rest of the work.

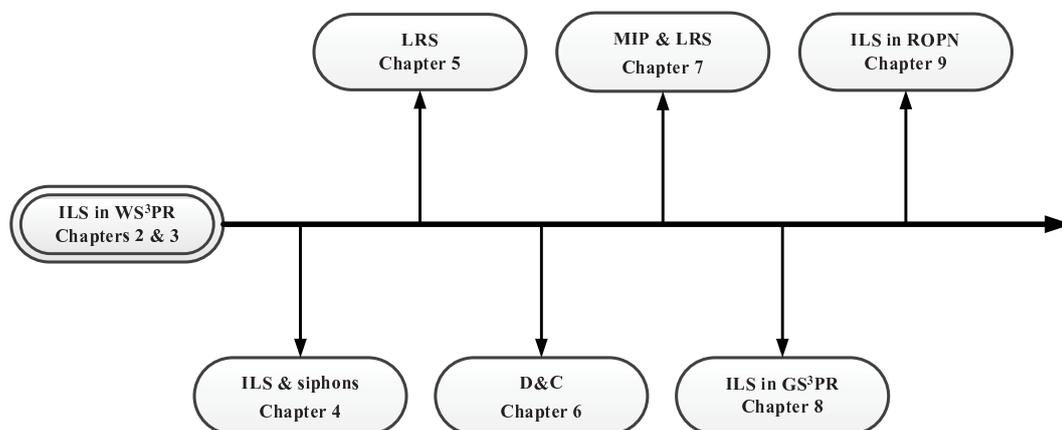


FIG. 6 – The framework of the thesis.

Chapter 1 introduces the basic definitions and notations of Petri nets used throughout this thesis.

Chapter 2 addresses the considered class of generalized Petri nets, WS^3PR , within the scope of

which the deadlock control and liveness-enforcement problems are investigated based on a kind of characteristic structures in subsequent chapters. Several important concepts are given. In addition, the CWs, CBs, and their relations with deadlocks in WS^3PR are elaborated with several examples.

Chapter 3 presents a new method based on ILSs of WS^3PR . The numerical relationships among weights, and between weights and initial markings are investigated based on simple circuits of resource places, which are the simplest structure of circular wait, rather than siphons. A WS^3PR satisfying a proposed restriction is inherently deadlock-free and live without adding monitors and arcs but only by configuring its initial markings.

Chapter 4 proposes a hybrid liveness-enforcing method for WS^3PR . The proposed method combines elementary siphons with an ILS-based method to prevent deadlocks and enforce liveness to the net class under consideration. The method addressed in Chapter 3 is further advanced in this chapter. It unveils and takes a full advantage of the ILSs of generalized Petri nets, which is hidden behind the arc weights, to achieve the liveness enforcement without any external control agent such as monitors. This hybrid method can identify and remove redundant monitors from a liveness-enforcing supervisor designed according to existing policies, improve the permissiveness, reduce the structural complexity of a controlled system, and consequently save the control implementation cost.

The work in Chapter 5 proposes a synthesis method of supervisors for flexible manufacturing systems modeled by WS^3PR . A concept of resource usage ratios (RU-ratios) is first presented to describe the occupation degree of a resource by an operation. Next, an ILS characterized by a special numerical relationship between arc-weights and initial markings is investigated from a perspective of RU-ratios. Then, a new kind of supervisors is synthesized on the ground of the generic nature of ILSs.

As a significant paradigm in computer science, the divide-and-conquer technique is investigated for enforcing liveness of WS^3PR modeling FMS in Chapter 6. In the dividing stage, a primary separation and an advanced one of the whole structure of a plant net model are successively presented. After the decomposition and simplification, the real niduses of non-liveness can be precisely located in the whole structure. In the conquering stage, a kind of liveness and resource usage ratio-

enforcing supervisors (LRSs) is adopted to conquer subnets without introducing new problematic structural objects. By a combination of all subnets, a live net model is gained, in which all siphons are minimally controlled.

Chapter 7 presents an iterative liveness-enforcing method for WS^3PR , which checks the liveness of net models using mixed integer programming (MIP) and controls the token allocations of resource places instead of siphons using LRSs.

Chapter 8 extends the ILS-base methods addressed in previous chapters to a more general class than WS^3PR based on an extended concept of WSDCs. The work in this chapter provides an insight into structures of generalized Petri nets and a new revenue to achieve the desired deadlock control.

In a comparative way, Chapter 9 discusses the intrinsically live structures in resource-oriented Petri nets based on those in process-oriented ones. The work in this chapter involves colored Petri net models to describe the operation route and the control mechanism.

Finally, Chapter 10 summarizes the thesis by highlighting the main contributions. Several open and interesting problems for future work are given.

0.3.1 Publications

This thesis includes the work from a number of published papers and technical reports, as follows :

Journal Papers : (published or accepted)

1. **Ding Liu**, ZhiWu Li, and MengChu Zhou, "A parameterized liveness and ratio-enforcing supervisor for a class of generalized Petri nets," *Automatica*, vol. 49, no. 11, pp.3167–3179, 2013.
2. **Ding Liu**, ZhiWu Li, and MengChu Zhou, "Hybrid liveness-enforcing policy for generalized Petri net models of flexible manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part A : Systems and Humans*, vol. 43, no. 1, pp. 85–97, 2013.
3. **Ding Liu**, ZhiWu Li, and MengChu Zhou, "Erratum to "Liveness of an extended S^3PR

- [*Automatica* 46 (2010) 1008-1018]”,” *Automatica*, vol. 48, no. 5, pp. 1003–1004, 2012.
4. **Ding Liu**, ZhiWu Li, and MengChu Zhou, “Liveness of an extended S^3PR ,” *Automatica*, vol. 46, no. 6, pp. 1008–1018, 2010.
 5. **Ding Liu**, ZhiWu Li, and MengChu Zhou, “Authors’ reply to ‘Comments on “Liveness of an extended S^3PR ” [Automatica 46(6) (2010) 1008–1018],” *Automatica*, vol. 50, no. 8, pp. 2201, 2014.
 6. ZhiWu Li and **Ding Liu**, “A correct minimal siphon extraction algorithm from a maximal unmarked siphon of a Petri net,” *International Journal of Production Research*, vol. 45, no. 9, pp. 2161–2165, 2007.
 7. YiFan Hou, Mi Zhao, and **Ding Liu**, “Deadlock control for a class of generalized Petri nets based on proper resource allocation,” *Asian Journal of Control*, vol. 17, no. 4, pp. 1-18, 2014. (DOI : 10.1002/asjc.1004)
 8. Mi Zhao, YiFan Hou, and **Ding Liu**, “Liveness-enforcing supervisors synthesis for a class of generalized Petri nets based on two-stage deadlock Control and mathematical programming,” *International Journal of Control*, vol. 83, no. 10, pp. 2053–2066, 2010.
 9. YiFan Hou, ZhiWu Li, Mi Zhao, and **Ding Liu**, “Extraction of elementary siphons in a class of generalized Petri nets using graph theory,” *Engineering Computations*, vol. 31, no. 2, pp. 331-352, 2014.
 10. YiFan Hou, ZhiWu, Li, Mi Zhao, and **Ding Liu**, “Extended elementary siphon-based deadlock prevention policy for a class of generalized Petri nets,” *International Journal of Computer Integrated Manufacturing*, vol. 27, no. 1, pp. 85-102, 2014.
 11. YiFan Hou, ZhiWu, Li, Mi Zhao, and **Ding Liu**, “Extension of elementary siphons in a class of generalized Petri net,” *Applied Mechanics and Materials-Innovation for Applied Science and Technology*, vol. 284-287, pp. 2238-2243, 2013.

Technical Report :

12. **Ding Liu**, ZhiWu Li, YiFan Hou, and MengChu Zhou, “Liveness-enforcement for generalized Petri nets using a divide-and-conquer paradigm,” Technical Report, 2014.

Conference Papers :

1. **Ding Liu**, YiFan Hou, Kamel Barkaoui, and MengChu Zhou, “Intrinsically Live Structures in Process and Resource-Oriented Petri Nets Modeling Automated Manufacturing Systems,” in *Proceedings of IEEE the 11th IEEE International Conference on Networking, Sensing, and Control*, pp. 578–583, Miami, FL, USA, Apr. 7-9, 2014.
2. **Ding Liu**, YiFan Hou, Kamel Barkaoui, and MengChu Zhou, “Liveness and Resource Usage Ratio-Enforcing Supervisor in a Class of Generalized Petri Nets,” in *Proceedings of the 10th IEEE ICCA*, pp. 1278-1283, Hangzhou, China, Jun. 12-14, 2013.
3. **Ding Liu**, Kamel Barkaoui, and MengChu Zhou, “On intrinsically live structure of a class of generalized Petri nets modeling FMS,” in *Proceedings of 11th International Workshop on Discrete Event Systems*, pp. 187–192, Guadalajara, Mexico, Oct. 3-5, 2012.
4. **Ding Liu**, Mi Zhao, HeSuan Hu, and AnRong Wang, “Hybrid liveness-enforcing method for Petri net models of flexible manufacturing systems,” in *Proceedings of IEEE International Conference on Mechatronics and Automation*, pp. 1813–1818, Xi’an, China, Aug. 4-7, 2010.
5. YiFan Hou, **Ding Liu**, and MengChu Zhou, “On iterative liveness-enforcement for a class of generalized Petri nets,” in *Proceedings of the 8th IEEE International Conference on Automation Science and Engineering*, pp. 184–189, Seoul, Korea, Aug. 20-24, 2012.
6. YuFeng Chen, **Ding Liu**, GaiYun Liu, and Kamel Barkaoui, “Computation of Resource Circuits of Petri Nets by Using Binary Decision Diagrams,” in *Proceedings of the 5th International Conference on Modeling, Simulation and Applied Optimization*, pp. 1–6, Hammamet, Tunisia, Apr. 28-30, 2013. (DOI : 10.1109/ICMSAO.2013.6552618)
7. YiFan Hou, **Ding Liu**, ZhiWu Li, and Mi Zhao, “Deadlock prevention using divide-and-conquer strategy for WS^3PR ,” in *Proceedings of IEEE International Conference on Mechatronics and Automation*, pp. 1635–1640, Xi’an, China, Aug. 4-7, 2010.
8. Mi Zhao, **Ding Liu**, and YiFan Hou, “Non-blocking supervisors for a class of discrete event systems based on two-stage deadlock control,” in *Proceedings of IEEE International Conference on Mechatronics and Automation*, pp. 1081–1086, Xi’an, China, Aug. 4-7, 2010.
9. ZhiWu Li, **Ding Liu**, and AnRong Wang, “A polynomial algorithm to decide the liveness for a class of Petri nets,” in *Proceedings of IEEE International Conference on Networking,*

0.3. THESIS ORGANIZATION

Sensing, and Control, pp. 255–259, Florida, USA, Apr. 23-26, 2006.

10. YiFan Hou, ZhiWu Li, **Ding Liu**, and Mi Zhao, “Elementary siphons in a class of generalized Petri nets and their application,” in *Proceedings of the 2011 International Conference on Advanced Mechatronics Systems*, pp. 103–108, Zhengzhou, China, Aug. 11-13, 2011.

Chapitre 1

Preliminaries of Petri Nets

This chapter gives some necessary concepts of Petri nets involved in the thesis. It is assumed that readers are familiar with the formalism of Petri nets. For more details, please refer to [61], [65], [66], and [72].

A generalized Petri net (structure) is a four-tuple $N = (P, T, F, W)$ where P and T are finite and nonempty sets. P is a set of places and T is a set of transitions with $P \cap T = \emptyset$. $F \subseteq (P \times T) \cup (T \times P)$ is called a flow relation, represented by arcs with arrows from places to transitions or from transitions to places. $W : (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$ is a mapping that assigns weights to arcs : $W(x, y) > 0$ if $(x, y) \in F$, and $W(x, y) = 0$ otherwise, where $x, y \in P \cup T$ and $\mathbb{N} = \{0, 1, 2, \dots\}$ is a set of non-negative integers. A net is said to be ordinary if $\forall (x, y) \in F, W(x, y) = 1$. A net $N = (P, T, F, W)$ is pure (self-loop free) if $W(x, y) > 0$ implies $W(y, x) = 0, \forall x, y \in P \cup T$. A pure net $N = (P, T, F, W)$ can be represented by its incidence matrix $[N]$, where $[N]$ is a $|P| \times |T|$ integer matrix with $[N](p, t) = W(t, p) - W(p, t)$.

Let $x \in P \cup T$ be a node of net N . The preset of x is defined as $\bullet x = \{y \in P \cup T | (y, x) \in F\}$. While the postset of x is defined as $x^\bullet = \{y \in P \cup T | (x, y) \in F\}$. $x^{\bullet\bullet} = \cup_{y \in x^\bullet} y^\bullet$ and $\bullet\bullet x = \cup_{y \in \bullet x} \bullet y$. The notations can be extended to a set of nodes as follows : given $X \subseteq P$ or $X \subseteq T$, $\bullet X = \cup_{x \in X} \bullet x$ and $X^\bullet = \cup_{x \in X} x^\bullet$. Given place p , we denote $\max\{W(p, t) | t \in p^\bullet\}$ (resp. $\min\{W(p, t) | t \in p^\bullet\}$) by \max_{p^\bullet} (resp. \min_{p^\bullet}). A sequence of nodes $x_1 \dots x_i \dots x_n$ is called a path of N if $\forall i \in \mathbb{N}_{n-1} = \{1, 2, \dots, n-1\}, x_{i+1} \in x_i^\bullet$, where $x_i \in P \cup T$. A simple one is a path from x_1 to x_n whose nodes are

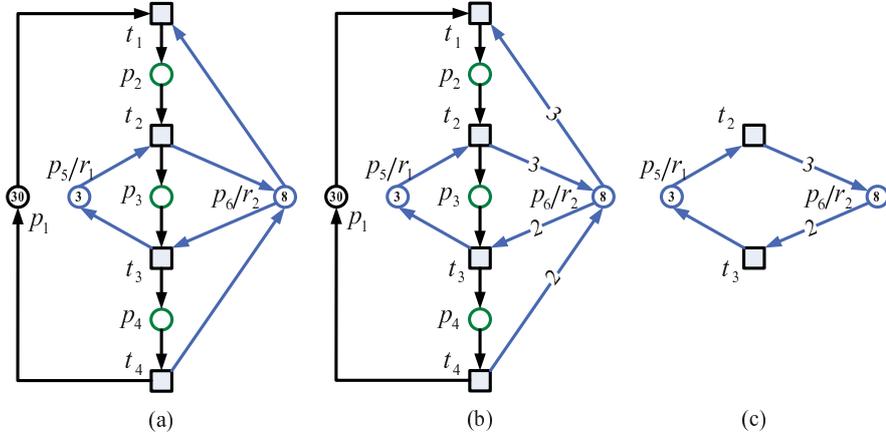


FIG. 1.1 – (a) An ordinary net, (b) a generalized net, and (c) a weighted digraph of resource places, G_R .

all different (except, perhaps, x_1 and x_n). It is called a simple circuit if $x_1 = x_n$.

A marking M of N is a mapping from P to \mathbb{N} . $M(p)$ denotes the number of tokens in place p . A place p is marked at M if $M(p) > 0$. It is insufficiently marked with respect to a transition $t \in p^\bullet$ at M if $M(p) < W(p, t)$. A subset $S \subseteq P$ is marked at M if at least one place in S is marked at M . The sum of tokens in all places in S is denoted by $M(S)$, i.e., $M(S) = \sum_{p \in S} M(p)$. S is said to be empty at M if $M(S) = 0$. S is insufficiently marked at M if $\forall p \in S, \forall t \in p^\bullet, M(p) < W(p, t)$. Conversely, it is sufficiently marked at M if $\exists p \in S, M(p) \geq \min_{p^\bullet}$. M_0 is called an initial marking of N and (N, M_0) is called a net system or marked net.

A transition $t \in T$ is enabled at marking M if $\forall p \in {}^\bullet t, M(p) \geq W(p, t)$. This fact is denoted as $M[t]$. Once a transition t fires, it gives a new marking M' . It is denoted as $M[t]M'$, where $M'(p) = M(p) - W(p, t) + W(t, p), \forall p \in P$. M' is reachable from M if there exist an enabled transition sequence $\delta = t_1 t_2 \cdots t_n$ and markings M_1, M_2, \dots, M_{n-1} such that $M[t_1]M_1[t_2]M_2 \cdots M_{n-1}[t_n]M'$ holds, which is denoted as $M[\delta]M'$ and satisfies the state equation $M' = M + [N]\vec{\delta}$, where $\vec{\delta} : T \rightarrow \mathbb{N}$ is a vector of non-negative integers, called a firing vector, and $\vec{\delta}(t)$ indicates the algebraic sum of all occurrences of t in δ . The set of all markings reachable from M by firing any possible sequence of transitions is denoted as $R(N, M)$. $R(N, M_0)$ is the set of reachable markings of a Petri net with the initial marking M_0 .

Let (N, M_0) be a net system with $N = (P, T, F, W)$. A transition $t \in T$ is live at M_0 if $\forall M \in R(N, M_0), \exists M' \in R(N, M), M'[t]$. (N, M) is live if $\forall t \in T, t$ is live at M . It is dead at M if $\nexists t \in T, M[t]$. It is deadlock-free (weakly live) if $\forall M \in R(N, M_0), \exists t \in T, M[t]$.

Definition 1.1 Let $N = (P, T, F, W)$ be a Petri net with $P_X \subseteq P$ and $T_X \subseteq T$. $N_X = (P_X, T_X, F_X, W_X)$ is called a subnet generated by $P_X \cup T_X$ such that $F_X = F \cap [(P_X \times T_X) \cup (T_X \times P_X)]$ and $\forall f \in F_X, W_X(f) = W(f)$.

Definition 1.2 Let (N_1, M_1) and (N_2, M_2) be two generalized Petri nets with $N_i = (P_i, T_i, F_i, W_i)$, $i \in \{1, 2\}$, where $P_1 \cap P_2 = P_C \neq \emptyset$ and $T_1 \cap T_2 = \emptyset$. (N, M) with $N = (P, T, F, W)$ is said to be their composition, denoted by $N_1 \circ N_2$, via the set of shared places P_C , if (1) $P = P_1 \cup P_2$, $T = T_1 \cup T_2$, $F = F_1 \cup F_2$, and $W = W_1 \cup W_2$; and (2) $\forall p \in P_1 \setminus P_C, M(p) = M_1(p)$; $\forall p \in P_2 \setminus P_C, M(p) = M_2(p)$; and $\forall p \in P_C, M(p) = \max\{M_1(p), M_2(p)\}$.

Definition 1.3 A P -vector is a column vector $I : P \rightarrow \mathbb{Z}$ indexed by P , where \mathbb{Z} is the set of integers. A P -vector I is said to be a P -invariant if $I \neq \mathbf{0}$ and $I^T[N] = \mathbf{0}^T$.

Definition 1.4 A P -invariant I is said to be a P -semiflow if every element of I is non-negative. $\|I\| = \{p | I(p) \neq 0\}$ is called the support of I . $\|I\|^+ = \{p | I(p) > 0\}$ denotes the positive support of P -invariant I and $\|I\|^- = \{p | I(p) < 0\}$ denotes the negative support of I . I is said to be a minimal P -invariant if $\|I\|$ is not a superset of the support of any other invariant and its components are mutually prime.

Definition 1.5 A multiset Ω , over a non-empty set D , is a mapping $\Omega : D \rightarrow \mathbb{N}$, denoted as a formal sum $\sum_{d \in D} \Omega(d)d$.

A P -invariant corresponds to a set of places whose weighted token count is a constant at any reachable marking. In the net shown in Fig. 1.1(b), there are three P -invariants in the expression of multiset, $I_1 = p_1 + p_2 + p_3 + p_4$, $I_2 = p_3 + p_5$, and $I_3 = 3p_2 + 2p_4 + p_6$.

A nonempty set $S \subseteq P$ is a siphon if $\bullet S \subseteq S^\bullet$. A siphon is minimal if there is no siphon contained in it as a proper subset. A minimal siphon S is said to be strict if $\bullet S \subset S^\bullet$. Siphons are a kind of significant structural objects extensively used in a large number of deadlock control and liveness-enforcing methods for ordinary and generalized Petri net models of FMS. For example, in the gene-

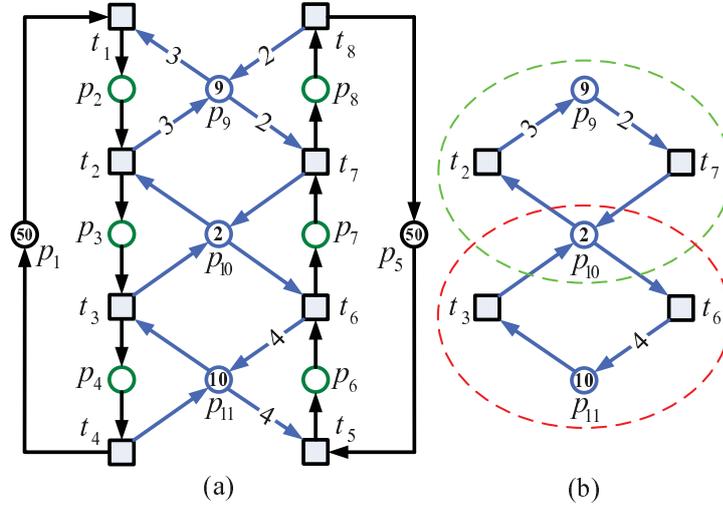


FIG. 1.2 – (a) A generalized net and (b) one of its subnets.

ralized Petri net models shown in Fig. 1.2(a), $S_1 = \{p_3, p_8, p_9, p_{10}\}$ and $S_2 = \{p_4, p_7, p_{10}, p_{11}\}$ are two strict minimal siphons. It is worth noting that a siphon does not carry the information of arc weights.

In Fig. 1.2(a), the example is a pure generalized Petri net model with the initial marking expressed in multiset $M_0 = 50p_1 + 50p_5 + 9p_9 + 2p_{10} + 10p_{11}$. Note that M_0 means that either of p_1 and p_5 has 50, p_9 has 9, p_{10} has 2, and p_{11} has 10 tokens, whereas the others have none. The net shown in Fig. 1.2(b) in one of its subnets as generated by $P_X = \{p_9, p_{10}, p_{11}\}$ and $T_X = \{t_2, t_3, t_6, t_7\}$.

Definition 1.6 Let (N, M_0) be a net system and S be a siphon of N . S is said to be max-marked (resp. min-marked) at a marking $M \in R(N, M_0)$ if $\exists p \in S$ such that $M(p) \geq \max_{p^\bullet}$ (resp. $M(p) \geq \min_{p^\bullet}$).

Definition 1.7 A siphon is said to be maximally (resp. minimally) controlled if it is max-marked (resp. min-marked) at any reachable marking.

A max-marked siphon implies that at least one place p in it contains enough tokens to fire any transition $t \in p^\bullet$. A min-marked one implies that at least one of its places p contains adequate tokens to fire at least one transition $t \in p^\bullet$. If a siphon is maximally controlled, it is also minimally controlled, but not vice versa. For example, $M = 41p_1 + 9p_4 + 50p_5 + 9p_9 + 2p_{10} +$

p_{11} is a reachable marking of the generalized Petri net model shown in Fig. 1.2(a). The siphon $S_2 = \{p_4, p_7, p_{10}, p_{11}\}$ is min-marked at this marking since $M(p_{11}) < \max_{p_{11}} \bullet$ and $M(p_{11}) \geq \min_{p_{11}} \bullet$. In fact, S_2 is min-marked at all reachable markings including the only one deadlock state, i.e., it is minimally controlled according to Definition 1.7.

Chapitre 2

Systems of Simple Sequential Processes with Weighted Resource Allocation

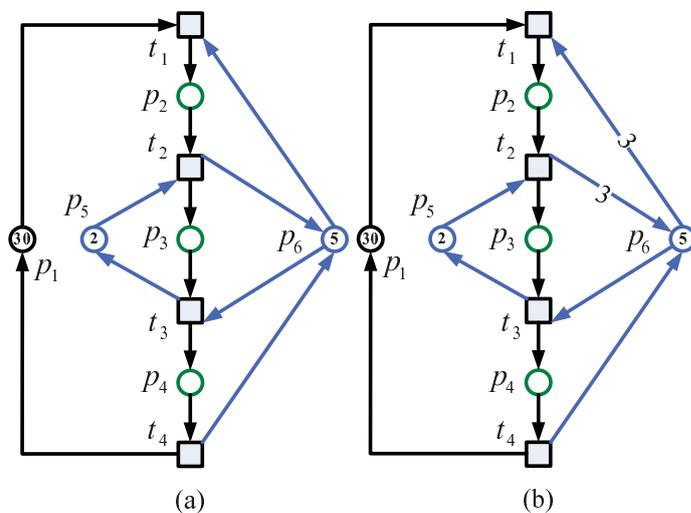
This chapter addresses the considered class of generalized Petri nets, the systems of simple sequential processes with weighted resource allocation (WS^3PR), within the scope of which the deadlock control and liveness-enforcing problems are investigated based on a kind of characteristic structures in subsequent chapters. Several important concepts are given, such as arc pairs, sufficient initial markings, and wait relations. In addition, the circular wait, circular blocking, and their relations with deadlocks are elaborated with several examples.

2.1 WS^3PR

The considered generalized Petri nets in the thesis, WS^3PR , is an extension to the systems of simple sequential processes with resources (S^3PR). S^3PR and WS^3PR are topologically isomorphic. The former is a subclass of the latter and recursively defined from a simple sequential process (S^2P) as follows.

Definition 2.1 A simple sequential process (S^2P) is an ordinary Petri net $N = (P_A \cup \{p_0\}, T, F)$ where

- (1) $P_A \neq \emptyset$ is called a set of activity or operation places ;
- (2) $p_0 \notin P_A$ is called an process idle place with $P_0 = \{p_0\}$; and


 FIG. 2.1 – (a) An S²PR (S³PR) net model and (b) a WS²PR (WS³PR) net model.

(3) N is a strongly connected state machine.

S²P represents a sequence of operations decided by a product specification. Every activity place in S²P needs one or more units of a single type of resources to complete an operation. An executing process with resources is modeled by S²PR or WS²PR.

Definition 2.2 A simple sequential process with resources (S²PR) is an ordinary Petri net $N = (P_A \cup P_0 \cup P_R, T, F)$ such that

- (1) The subnet generated by $X = P_A \cup P_0 \cup T$ is an S²P;
- (2) $P_R \neq \emptyset$ is called a set of resource places with $(P_A \cup P_0) \cap P_R = \emptyset$;
- (3) $\forall p \in P_A, \forall t \in \bullet p, \forall t' \in p \bullet, \exists r_p \in P_R, \bullet t \cap P_R = t' \bullet \cap P_R = \{r_p\}$;
- (4) The following statements are verified : (a) $\forall r \in P_R, \bullet \bullet r \cap P_A = r \bullet \bullet \cap P_A \neq \emptyset$ and (b) $\forall r \in P_R, \bullet r \cap r \bullet = \emptyset$; and
- (5) $\bullet \bullet (P_0) \cap P_R = (P_0) \bullet \bullet \cap P_R = \emptyset$.

An S²PR model represents a situation in which each operation in the process is associated with only one resource, and the acquisition and release of a resource by an operation is a pair of transitions. The original net shown in Fig. 2.1(a) is an S²PR. The activity place p_3 exclusively uses resource place p_5 , while activity places p_2 and p_4 share resource place p_6 .

2.1. WS³PR

Definition 2.3 A simple sequential process with weighted resources allocation (WS²PR), is a generalized Petri net $N = (P_A \cup P_0 \cup P_R, T, F, W)$ such that

- (1) The subnet generated by $X = P_A \cup P_0 \cup T$ is an S²P;
- (2) $P_R \neq \emptyset$ and $(P_A \cup P_0) \cap P_R = \emptyset$;
- (3) $\forall p \in P_A, \forall t \in \bullet p, \forall t' \in p^\bullet, \exists r_p \in P_R, \bullet t \cap P_R = t'^\bullet \cap P_R = \{r_p\}$;
- (4) The two following statements are verified : (a) $\forall r \in P_R, \bullet\bullet r \cap P_A = r^{\bullet\bullet} \cap P_A \neq \emptyset$, and (b) $\forall r \in P_R, \bullet r \cap r^\bullet = \emptyset$;
- (5) $\bullet\bullet(P_0) \cap P_R = (P_0)^{\bullet\bullet} \cap P_R = \emptyset$; and
- (6) The three statements are true : (a) $\forall x, y \in P_A \cup P_0 \cup T, W(x, y) = 1$ if $(x, y) \in F$, (b) $\forall x, y \in P_R \cup T, W(x, y) \geq 1$ if $(x, y) \in F$, and (c) $W(r, t) = W(t', r)$, where $r \in P_R, t, t' \in T, \bullet t \cap P_R = t'^\bullet \cap P_R = \{r\}$.

According to the definition of WS²PR, every resource place is used by at least one operation place and every operation place uses only one resource place. Two arcs of any arc pair having the same weight guarantee that tokens are neither destroyed nor created, i.e., conservativeness. Fig. 2.1(b) gives a WS²PR. Obviously, S²PR is a subclass of WS²PR.

Definition 2.4 Given a WS²PR N , for $r \in P_R, H(r) = (\bullet\bullet r \cap P_A) \cup (r^{\bullet\bullet} \cap P_A)$ is called a set of holders of r .

A holder $h(r) \in H(r)$ is an activity place that completes an operation by requesting and holding some tokens of resource place r . It is said to be shared if $|H(r)| > 1$. For example, in Fig. 2.2(a), $H(p_{11}) = \{p_4, p_{10}\}$. p_4 and p_{10} need one and four tokens in p_{11} every time to complete an operation, respectively. The following is an alternative definition of the set of holders, which is expressed in multisets. Essentially, it is equivalent to Definition 2.4.

Definition 2.5 Let r be a resource place and S be a strict minimal siphon in an S²PR. The set of holders of r is defined as the difference of two multisets $H(r) = I_r - r$, where I_r is a unique minimal P -semiflow $I_r \in \mathbb{N}^{|P|}$ associated with resource place r such that $\{r\} = \|I_r\| \cap P_R, P_0 \cap \|I_r\| = \emptyset, P_A \cap \|I_r\| \neq \emptyset$, and $I_r(r) = 1$.

Definition 2.6 $\bar{S} = \sum_{r \in S^R} H(r) - \sum_{r \in S^R, p \in S^A} I_r(p)p$ is called the complementary set of siphon S , where $S^R = S \cap P_R$ and $S^A = S \cap P_A$.

For example, in Fig. 2.1(b), $S = \{p_4, p_5, p_6\}$, $I_{p_5} = p_3 + p_5$, $I_{p_6} = 3p_2 + p_4 + p_6$, $H(p_5) = p_3$, $H(p_6) = 3p_2 + p_4$, and $\bar{S} = (3p_2 + p_3 + p_4) - (p_4) = 3p_2 + p_3$.

Definition 2.7 An arc pair is a pair of arcs, (t_i, r) and (r, t_j) , where $r \in P_R$, $\bullet t_i \cap \bullet t_j \cap P_A = \{p\}$, and $p \in H(r)$. (t_i, r) is the paired arc of (r, t_j) and vice versa.

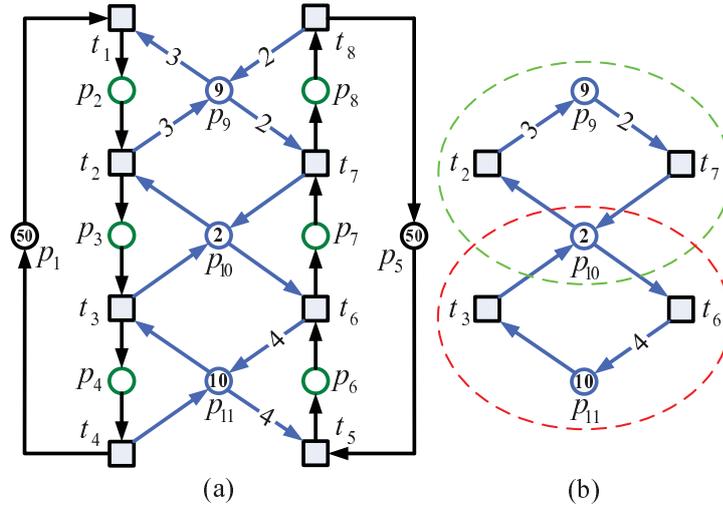


FIG. 2.2 – (a) A generalized net and (b) one of its subnets.

An arc from a resource place to a transition represents that an activity place acquires one or more tokens from the resource place. An arc from a transition to a resource place represents that an activity place gives back one or more tokens to the resource place. Two arcs in an arc pair modeling the behavior of a holder $h(r)$ acquiring and releasing tokens from resource place r are of the same weight to guarantee the resource conservativeness. In Fig. 2.2(a), (t_2, p_9) and (p_9, t_1) , (t_8, p_9) and (p_9, t_7) , (t_3, p_{10}) and (p_{10}, t_2) , (t_7, p_{10}) and (p_{10}, t_6) , (t_4, p_{11}) and (p_{11}, t_3) , and (t_6, p_{11}) and (p_{11}, t_5) are six arc pairs. Moreover, $W(t_2, p_9) = W(p_9, t_1) = 3$, $W(t_8, p_9) = W(p_9, t_7) = 2$, $W(t_3, p_{10}) = W(p_{10}, t_2) = 1$, $W(t_7, p_{10}) = W(p_{10}, t_6) = 1$, $W(t_4, p_{11}) = W(p_{11}, t_3) = 1$, and $W(t_6, p_{11}) = W(p_{11}, t_5) = 4$.

If $t \in P_0^\bullet$ (resp. $t \in \bullet P_0$), t is called a source (resp. sink) transition. Note that weights of arcs connecting activity and process idle places with transitions in a WS²PR are all equal to one, whereas those of arcs connecting resource places are equal to or greater than one.

2.1. WS³PR

Let $N = (P_A \cup P_0 \cup P_R, T, F, W)$ be a WS²PR. An initial marking M_0 is said to be acceptable if : (1) $\forall p_0 \in P_0, M_0(p_0) \geq 1$; (2) $\forall p \in P_A, M_0(p) = 0$; and (3) $\forall r \in P_R, \forall t \in r^\bullet, M_0(r) \geq W(r, t)$. An acceptable initial marking is said to be abundant if $\forall r \in P_R, \forall t \in r^\bullet, M_0(r) \geq \sum_{t \in r^\bullet} W(r, t)$. An acceptable but not abundant initial marking implies that every holder of the resource place has a chance to use tokens in the resource place while the others may have to wait. An abundant one implies that it is possible that every holder can obtain adequate tokens from the resource place at the same time.

Definition 2.8 A system of S²PR, called S³PR for short, is defined recursively as follows :

- (1) An S²PR is an S³PR ; and
- (2) Let $N_i = (P_{A_i} \cup P_{0_i}, T_i, F_i), i \in \{1, 2\}$, be two S³PR such that $(P_{A_1} \cup P_{0_1}) \cap (P_{A_2} \cup P_{0_2}) = \emptyset, P_{R_1} \cap P_{R_2} = P_C \neq \emptyset$, and $T_1 \cap T_2 = \emptyset$. Then, the net $N = (P_A \cup P_0 \cup P_R, T, F)$ resulting from the composition of N_1 and N_2 via P_C (denoted as $N_1 \circ N_2$) such that (a) $P_A = P_{A_1} \cup P_{A_2}$, (b) $P_0 = P_{0_1} \cup P_{0_2}$, (c) $P_R = P_{R_1} \cup P_{R_2}$, and (d) $T = T_1 \cup T_2$, and $F = F_1 \cup F_2$ is also an S³PR.

S³PR is widely accepted and used to model and analyze deadlock control problems in literature. The classic problem of dining philosophers, as shown in Fig. 2.3, is modeled with S³PR. Next, simple sequential process with weighted resource allocation, WS²PR for short, is defined as the same as S²PR except for some arcs' multiplicity.

Definition 2.9 A system of WS²PR, called WS³PR for short, is defined recursively as follows :

- (1) A WS²PR is a WS³PR ; and
- (2) Let $N_i = (P_{A_i} \cup P_{0_i} \cup P_{R_i}, T_i, F_i, W_i), i \in \{1, 2\}$, be two WS³PR such that $(P_{A_1} \cup P_{0_1}) \cap (P_{A_2} \cup P_{0_2}) = \emptyset, P_{R_1} \cap P_{R_2} = P_C \neq \emptyset$, and $T_1 \cap T_2 = \emptyset$. Then, the net $N = (P_A \cup P_0 \cup P_R, T, F, W)$ resulting from the composition of N_1 and N_2 via P_C (denoted by $N_1 \circ N_2$) such that (a) $P = P_{A_1} \cup P_{A_2}$, (b) $P_0 = P_{0_1} \cup P_{0_2}$, (c) $P_R = P_{R_1} \cup P_{R_2}$, (d) $T = T_1 \cup T_2$, (e) $F = F_1 \cup F_2$, and (f) $W = W_1 \cup W_2$ is also a WS³PR.

The net model shown in Fig. 2.2(a) is a WS³PR with an acceptable initial marking $M_0 = 50p_1 + 50p_5 + 9p_9 + 2p_{10} + 10p_{11}$. If all weights of this WS³PR net are changed into one, an isomorphic S³PR is obtained. Hence, S³PR is a subclass of WS³PR.

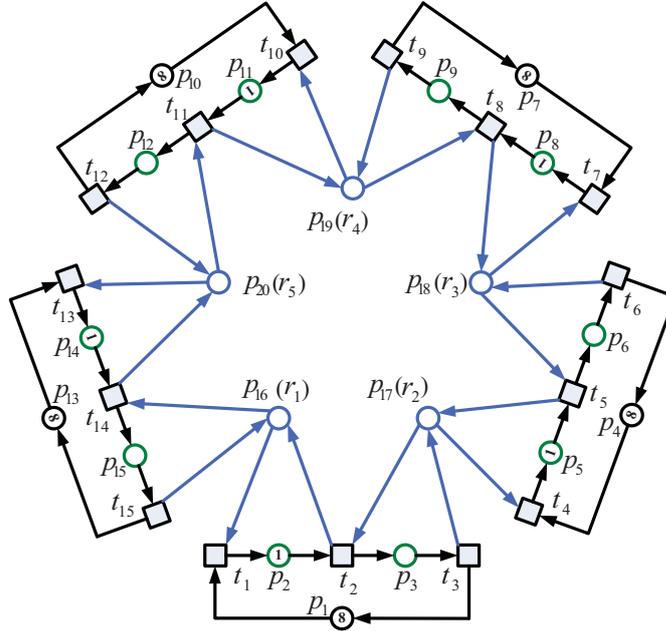


FIG. 2.3 – An S³PR model for the problem of Dining Philosophers.

2.2 Circular wait and circular blocking in WS³PR

In FMS, when a deadlock occurs, Coffman's four necessary conditions all hold. The first three, no preemption (i.e. a resource cannot be released from a job until it is complete), mutual exclusion (each resource can be used for only one job at a time), and hold while waiting (a job holds its resources while waiting for the part to move into the next job) are always satisfied in manufacturing practice. Only the last one, circular wait condition (two or more processes form a circular chain where each process waits for a resource that next process in the chain holds) can be prevented through various methods to make the whole system deadlock-free. Similar to the structure of circular wait and state of circular blocking studied in [52], their definitions and applications in the context of WS³PR for the purpose of deadlock analysis are presented next. The weights of arcs have to be considered in the efforts to prevent the state of circular blocking.

Definition 2.10 A place p is said to be dead with respect to M if $M(p) < \min_{t \in p \bullet} [W(p, t)]$ and there exists no $M' \in R(N, M)$ such that $M'(p) > M(p)$. A transition t is said to be dead with respect to M if there exists no $M' \in R(N, M)$ that enables t .

Definition 2.11 Given a marked WS³PR, a part path in the net model is said to be deadlocked if there is a dead transition on that part path.

Lemma 2.1

Given a marked WS³PR, there is a deadlocked part path if and only if there is a dead resource place in it.

Proof : (*if*) If resource place r is dead, then $m(r) < \min_{t \in p \bullet} [W(r, t)]$ and all the transitions in $\bullet r$ are dead. Obviously $\bullet r = H(r) \bullet$, where $H(r) = (\bullet \bullet r \cap P_A) \cup (r \bullet \bullet \cap P_A)$. This means a deadlocked part path. (*only if*) It follows from Definition 2.11 : if there exists a deadlocked part path, then there is on that part path a dead transition, t . It can be dead only if its input resource place in $\bullet t \cap P_R$ is dead.

A deadlocked part path means that the operating process cannot continue due to lack of its needed resources. A system deadlock is composed of some deadlocked part paths. In other words, there is no system deadlock if no part path becomes deadlocked. For example, Fig. 2.1(b) has only one part path that can never be deadlocked under the initial marking $M_0 = 30p_1 + 2p_5 + 5p_6$.

Definition 2.12 A resource place $r \in P_R$ is called a shared resource place if $|H(r)| \geq 2$ and a non-shared one if $|H(r)| = 1$.

The competition for shared resources is the very origin of deadlock. In WS³PR, there is no deadlock if every resource place is exclusively used by one holder, i.e., they are all non-shared.

A method proposed in the thesis deals with the liveness and ratio-enforcing problems within the scope of WS³PR by using the concepts of wait relations and circular waits defined as follows. For any two resource places $r_i, r_j \in P_R$, r_i is said to wait for r_j , denoted by $r_j \xrightarrow{t} r_i$, if the availability of r_j is an immediate requirement for the release of r_i by firing transition t , or equivalently, $\exists t \in \bullet r_i \cap r_j \bullet$. An R -path between r_i and r_k is defined as a set of resource places and transitions such that $r_k \xrightarrow{t_l} r_j \xrightarrow{t_m} \dots \xrightarrow{t_n} r_i$. Then r_i is said to wait over an R -path for r_k , denoted by $r_k \xrightarrow{t_l \dots t_n} r_i$, if there is an R -path between r_i and r_k .

Definition 2.13 A set of nodes $C = R^C \cup T^C$ in a WS³PR is said to be a circular wait (CW) such that for any ordered pair $\{r_i, r_j\} \subseteq R^C$, $r_i \xrightarrow{t_l \dots t_n} r_j$, where $R^C \subseteq P_R$ and $T^C \subseteq T$ with $|R^C| > 1$, $|T^C| > 1$, and $t_l, \dots, t_n \in T^C$.

2.2. CIRCULAR WAIT AND CIRCULAR BLOCKING IN WS³PR

A simple CW is a set of nodes $C = R^C \cup T^C$ such that, for some appropriate relabeling one has $r_1 \xrightarrow{t_1} r_2 \xrightarrow{t_2} \dots \xrightarrow{t_{q-1}} r_q \xrightarrow{t_q} r_1$, with $r_i \neq r_j$ for $i \neq j$, $1 \leq i, j \leq q$. In the subnet shown in Fig. 2.2(b), $C_1 = \{t_2, p_9, t_7, p_{10}, t_6, p_{11}, t_3\}$, $C_2 = \{t_2, p_9, t_7, p_{10}\}$, and $C_3 = \{t_3, p_{10}, t_6, p_{11}\}$ are three CWs contained in the WS³PR. C_2 and C_3 are two simple ones.

Proposition 2.1

A WS³PR $N = (P_A \cup P_0 \cup P_R, T, F, W)$ is live if no circular wait is formed by P_R .

Proof : Obviously, there is no structure to satisfy the last one of Coffman's four necessary conditions.

Given a simple CW $C = \{r_1, t_1, r_2, t_2, \dots, r_i, t_i\}$, we can partition $\bullet r_i$ into $\bullet r_i = \bullet r_{io} \cup \bullet r_{i+}$, where $\bullet r_{io} = T^C$, the set of input transitions of r_i with input arcs from any other $r_j \in C$, and $\bullet r_{i+} = \{t \in T \mid \bullet t \cap R^C = \emptyset\}$, the set of input transitions of r_i without input arcs from any other $r_j \in C$. Let $H(C) = \bigcup_{i=1}^n H(r_i)$, $r_i \in R^C$ and partition it into $H(C) = H_o(C) \cup H_+(C)$, where $H_o(C) = \{p \in H(C) \mid p^\bullet \subseteq \bullet r_{io}, r_i \in R^C\}$ and $H_+(C) = \{p \in H(C) \mid p^\bullet \subseteq \bullet r_{i+}, r_i \in R^C\}$. It is worth noting that all the places in $H_+(C)$ are holders of the shared resources contained in C only. For example, in the aforementioned circular wait $C_3 = \{t_3, p_{10}, t_6, p_{11}\}$, we have $H(C) = \{p_3, p_4, p_6, p_7\}$, $\bullet r_{io} = \{t_3, t_6\}$, $\bullet r_{i+} = \{t_4, t_7\}$, $H_o(C) = \{p_3, p_6\}$, and $H_+(C) = \{p_4, p_7\}$. As proved in [52], the places in $R^C \cup H_+(C)$ form a minimal siphon S_C . This result still holds in WS³PR since the definition of a siphon has nothing to do with the weights of its related arcs.

Definition 2.14 A circular blocking (CB) in a WS³PR (N, M_0) with $N = (P, T, F, W)$ is a state $M \in R(N, M_0)$ such that :

- (1) $\forall r_i \in C, \forall t \in r_i^\bullet \cap C, M(r_i) < W(r_i, t)$; and
- (2) for each $r_i \in C, \forall p \in H(r_i)$ with $M(p) \neq 0, p \in H_o(C)$.

In this case, C is said to be in a CB.

A CB is a necessary condition of deadlocks. In the WS³PR net shown in Fig. 2.2(a), the circular wait $C_2 = \{t_2, p_9, t_7, p_{10}\}$ falls into a CB at marking $M = 47p_1 + 3p_2 + 46p_5 + 2p_6 + 2p_7 + 2p_{11}$.

Lemma 2.2

Given a marked WS³PR, if a circular wait C is in a CB, then all resource places in C are dead.

Proof : The proof is similar to the one in [52] with a minor modification for generalization. Condition (ii) of Definition 2.14 implies that any $r_i \in C$ has one or more tokens if and only if some $r_j \in C$, $j \neq i$, can have a token. However, by condition (i) all the resource places in C are insufficiently marked, and hence none of them can ever receive any token, which means that they are all dead.

Lemma 2.3

Given WS³PR, $r \in P_R$ with $m(r) < \min_{t \in r^\bullet} [W(r, t)]$ is dead if and only if either of the following holds.

- (1) \exists CW $C \subset P_R \ni r \in C$ and C is in CB ; and
- (2) For each $p \in H(r)$ with $m(p) \neq 0$, \exists CW $C \subset P_R$ and $r^* \in C \ni \bullet(p^\bullet) \cap P_R \hookrightarrow r^*$, with all the resources on the resource path being dead and C being in CB.

Proof : (if) Sufficiency follows from Lemma 2.2. (only if) Let r be dead and let $p \in H(r)$ such that $m(p) \neq 0$, i.e., p currently holds at least one of the tokens of r . Since r is dead, then $p^\bullet(\subseteq \bullet r)$ and $\bullet(p^\bullet) \cap R(\subseteq \bullet\bullet r \cap R)$ must be dead. Iterating this process means at least one circular wait in CB.

Lemma 2.3 states that the existence of a dead resource place is equivalent to the existence of a state of CB in a WS³PR from a structural viewpoint.

Lemma 2.4

Given WS³PR, there exists a CB if and only if there is a deadlocked part path.

Proof : It follows from Lemmas 2.1 and 2.3.

This lemma establishes the equivalence between a CB and a deadlocked part path. For example, the S³PR with only one part path in Fig. 2.1(a) is deadlocked when the simple circuit $C = \{p_5, p_6\}$ is in CB. Compared with S³PR in Fig. 2.1(a), the only part path of the WS³PR in Fig. 2.1(b) is never deadlocked since the simple circuit of resource places $C = \{p_5, p_6\}$ is never in CB of which the reason will be explained later.

Theorem 2.1

Given a marked WS³PR, it is live if and only if there is no CW in CB.

Proof : This follows from Lemma 2.4. There will never be a deadlocked part path in the net model. Hence, the marked WS³PR (N, M_0) is deadlock-free and live.

2.2. CIRCULAR WAIT AND CIRCULAR BLOCKING IN WS^3PR

Here, with an attempt to avoid a siphon-based method, one analyzes the structure of CW, the state of CB, deadlocked part path, and resulting deadlock. Being different from ordinary Petri nets, arc weights must be considered during these analyses in WS^3PR . The numerical relationship between weights and initial marking affects the liveness of a WS^3PR as well as the structure of circular wait.

Chapitre 3

Intrinsically Live Structure

Most existing prevention methods tackle the deadlock issue arising in flexible manufacturing systems modeled with Petri nets by adding monitors and arcs. Instead, this chapter presents a new one based on a characteristic structure of a system of simple sequential processes with weighted resource allocation (WS^3PR), an extension of system of simple sequential processes with resources (S^3PR) with weighted arcs. The numerical relationships among weights, and between weights and initial markings are investigated based on simple circuits of resource places, which are the simplest structure of circular wait, rather than siphons. A WS^3PR satisfying a proposed restriction is inherently deadlock-free and live by configuring its initial markings. A set of polynomial algorithms are developed to implement the proposed method.

3.1 Introduction

The study in this chapter deals with WS^3PR , a weighted extension of S^3PR , which is a subclass of WS^3PSR , S^3PGR^2 , or S^4R . Being different from ordinary Petri nets, the role of weights of arcs in determining the liveness of general ones should be carefully considered. Weights of arcs in WS^3PR mean a multiple requirement for resources of an operation. The token counts in resource places and numerical relationship between markings and weights restrict the allocation of system resources. Hence, the studies of structures including arc weights and dynamic properties should be combined together during the evolution analysis of a WS^3PR . Generally, it is believed

that deadlocks are caused by lack of system resources or improper resource allocation. In some situations, the proposed method can make a system live by decreasing system resources instead of simply increasing them. A proper resource allocation is practically guaranteed by a proper numerical relationship between initial markings of resource places and arc weights. Compared with most existing deadlock prevention control policies, the new one does not add any monitors and arcs. The control cost of software and hardware can be saved. Meanwhile, the problem of structural complexity can be avoided. The proposed method is based on a characteristic structure of WS³PR. Before any deadlock prevention policy is applied, its structure should be checked to decide whether it is inherently deadlock-free and live under certain initial markings. The relation between weights and initial markings is established by using simple circuits of resource places. Polynomial algorithms are developed to implement the proposed method.

3.2 Intrinsically live structure

The results in Chapter 2 show the equivalence between a deadlock and a circular wait in circular blocking in WS³PR. In the following, the proposed restrictions on weights and initial marking of resource places can ensure live WS³PR.

Definition 3.1 Given a WS³PR $N = (P_A \cup P_0 \cup P_R, T, F, W)$, a weighted digraph of resource places, denoted as $G_R = (P'_R, T_R, F_R, W_R)$, is a subnet of N such that :

- (1) $P'_R \subseteq P_R$;
- (2) $T_R = \bullet P'_R \cap P'_R \bullet$;
- (3) $F_R = [(P'_R \times T_R) \cup (T_R \times P'_R)] \cap F$; and
- (4) $W_R = W(x, y), \forall x, y \in P'_R \cup T_R$.

For example, Fig. 3.1(b) shows a weighted digraph of resource places derived from the WS³PR in Fig. 3.1(a). An algorithm with which a weighted digraph of resource places can be obtained from a WS³PR is given in the subsequent section. In this section, an intrinsically live structure (ILS) of WS³PR and a corresponding liveness-enforcing method are presented with explanatory examples.

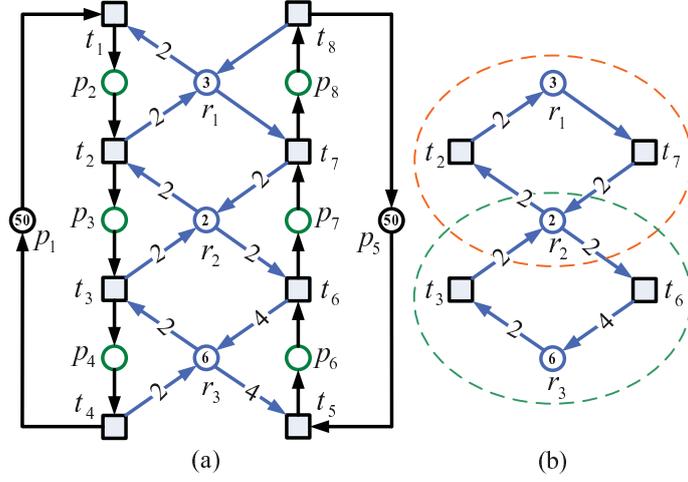


FIG. 3.1 – (a) A marked WS³PR net model and (b) a corresponding weighted digraph of resource places.

Originally inspired by [52], a generalized circular wait (CW), instead of siphons, is employed to investigate deadlocks and the liveness property in the previous chapter. We attempt to explore the relationship between the liveness and structure of WS³PR with an object that is simpler than siphons.

Definition 3.2 A weighted simple directed circuit (WSDC), denoted by $c_W = (P_R^C, T_R^C, F_R^C, W_R^C)$, is a simple circuit derived from a weighted digraph $G_R = (P'_R, T_R, F_R, W_R)$ of resource places such that :

- (1) $P_R^C = \{r_1, r_2, \dots, r_m\} \subseteq P'_R$, $T_R^C = \{t_1, t_2, \dots, t_m\} \subseteq T_R$ ($m \geq 2$), $r_1^\bullet \cap {}^\bullet r_2 = \{t_2\}$, $r_2^\bullet \cap {}^\bullet r_3 = \{t_3\}$, ..., and $r_m^\bullet \cap {}^\bullet r_1 = \{t_1\}$;
- (2) $F_R^C = [(P_R^C \times T_R^C) \cup (T_R^C \times P_R^C)] \cap F_R$; and
- (3) $W_R^C = W(x, y), \forall x, y \in P_R^C \cup T_R^C$.

For simplicity, we use $c_W = t_1 r_1 t_2 r_2 t_3 \dots t_n r_n t_1$ to denote a WSDC. In Fig. 3.1(b), there are two WSDCs, $c_{W1} = t_2 r_1 t_7 r_2 t_2$ and $c_{W2} = t_3 r_2 t_6 r_3 t_3$. A WSDC is an augmented simple CW. The difference between a WSDC and a simple CW is that a WSDC contains arcs and their assigned weights but a simple CW does not. A simple CW derived from a WSDC is denoted as $C = [c_W] = P_R^C \cup T_R^C$. For example, a derived simple CW from c_{W1} is $C_1 = [c_{W1}] = \{t_2, r_1, t_7, r_2\}$.

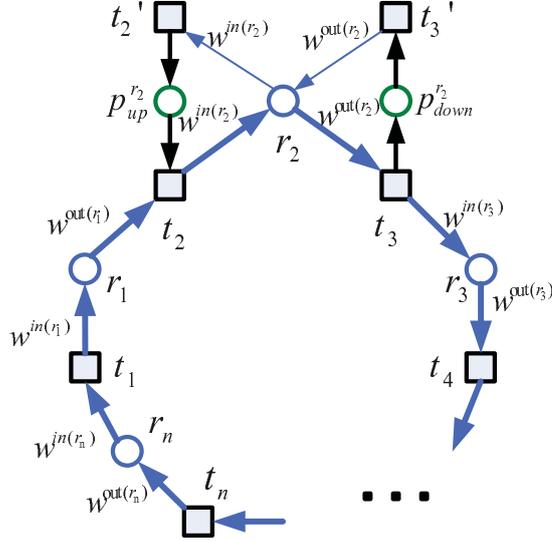


FIG. 3.2 – A general WSDC.

A general WSDC is shown in Fig. 3.2. A WSDC represents a circular competition relationship among multiple activity places jointly using resource places in it. In order to break a circle, one just needs break a chain of the circle. A more fundamental structural unit (competition path) is derived from WSDC and used to depict a competition relation between a pair-wise activity places as follows. Please note that the sequence of analysis intrinsically live structures here is from a WSDC to a competition path and further to the resource place of the competition path. In Chapter 5, the construction of an intrinsically live structure is in the reserve sequence.

Proposition 3.1

In WS^3PR , a WSDC must contain at least one shared resource place.

Proof : Proof is done by contradiction. Without loss of generality, let us consider a circuit with 4 vertices, i.e. r_1, r_2, t_1 , and t_2 . Both r_1 and r_2 are non-shared resource places. Suppose that p_1 and p_2 are their holders, respectively. Then, $\bullet\bullet p_1 = p_1 \bullet\bullet = p_2$ and $\bullet\bullet p_2 = p_2 \bullet\bullet = p_1$. This contradicts the definition of WS^3PR .

Definition 3.3 A simple path $l = (P_R^L, T_R^L, F_R^L, W_R^L)$ is called a resource path derived from $G_R = (P'_R, T_R, F_R, W_R)$ such that :

- (1) $P_R^L = \{r\} \subseteq P'_R$;

3.2. INTRINSICALLY LIVE STRUCTURE

- (2) $T_R^L = \{t_t, t_h\}$, where there exist a tail transition t_t with $\{t_t\} = \bullet r \cap T_R$, and a head one t_h with $\{t_h\} = r \bullet \cap T_R$;
- (3) $F_R^L = \{(t_t, r), (r, t_h)\} \subseteq F$; and
- (4) $W_R^L = \{W(t_t, r), W(r, t_h)\} \subseteq W_R$; where (t_t, r) and (r, t_h) are called the in-arc and out-arc of a resource path, respectively. If the in-arc and out-arc of a resource path are not an arc-pair, this resource path is called a competition path, denoted as l_{cp} .

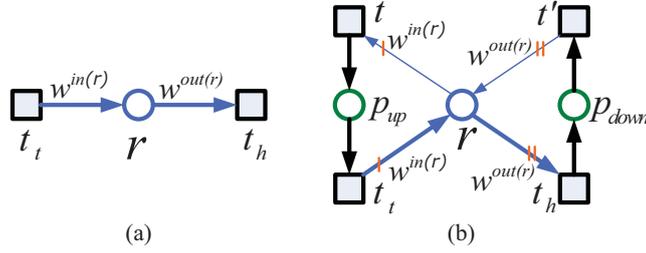


FIG. 3.3 – (a) A competition path and (b) a subnet containing the competition path and its upstream and downstream activity places.

For simplicity, a resource/competition path as shown in Fig. 3.3(a) is denoted by $t_t r t_h$. The weights of an in-arc and an out-arc are denoted by $w^{in(r)}$ and $w^{out(r)}$, respectively. We use \mathbf{L} to denote a set of all resource paths found in a weighted digraph of resource places G_R , i.e., $\mathbf{L} = \bigcup \{l_i\}$. $\mathbf{L}_{CP} = \bigcup \{l_{cp_i}\}$ denotes a set of all competition paths derived from G_R , and $\mathbf{L}_W = \{l_{cp} = t_t r t_h | l_{cp} \in \mathbf{L}_{CP}, W(t_t, r) > W(r, t_h)\}$. According to the definition of a competition path $t_t r t_h$, resource place r in it is shared, i.e., it has at least two holders. However, a competition path depicts a competition relationship between two holders of r only. As shown in Fig. 3.3(b), an upstream (resp. downstream) activity place p_{up} (resp. p_{down}) of a competition path $l_{cp} = t_t r t_h$ is a holder of resource place r with $t_t \in p_{up}^\bullet$ (resp. $t_h \in \bullet p_{down}$). (t_t, r) and (r, t) are an arc-pair with the same weight $w^{in(r)}$, whereas (r, t_h) and (t', r) are so with $w^{out(r)}$. A competition path has only one pair of determined upstream and downstream activity places. Note that the downstream activity place of a competition path $l_{cp} = t_t r t_h$ always belongs to $H_+([c_W])$, where $l_{cp} = t_t r t_h$ is contained in $WSDC c_W$. In addition, there always exists at least one downstream activity place belonging to $H_+(C)$ with respect to a circular wait C that may contain several simple circular waits. A competition path shows a relationship of competition for the same resource between an upstream activity place

3.2. INTRINSICALLY LIVE STRUCTURE

and a downstream one represented by an in-arc and an out-arc of the resource place, respectively.

For example, as shown in Fig. 3.1(b), $\mathbf{L} = \{t_2r_1t_7, t_7r_2t_2, t_3r_2t_2, t_7r_2t_6, t_3r_2t_6, t_6r_3t_3\}$, $\mathbf{L}_{CP} = \{t_2r_1t_7, t_7r_2t_2, t_3r_2t_6, t_6r_3t_3\}$, and $\mathbf{L}_W = \{t_2r_1t_7, t_6r_3t_3\}$. Take $l_{cp} = t_2r_1t_7$ as an example. The weight of l_{cp} 's in-arc (resp. out-arc) represents a specific requirement of tokens in resource place r_1 by upstream (resp. downstream) activity place p_2 (resp. p_8).

Proposition 3.2

There exists at least one competition path in a WSDC.

Proof : This follows from Definitions 3.2 and 3.3, and Proposition 3.1.

Theorem 3.1

A WS^3PR N is live if there is no WSDC in the weighted digraph G_R derived from N .

Proof : It is true since no structure of circular wait exists in the net.

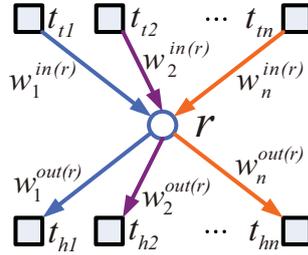


FIG. 3.4 – Multiple competition paths with a same resource place r .

A single competition path $l_{cp} = t_t r t_h$ represents a competitive relationship between an upstream activity place and a downstream activity place of l_{cp} jointly using resource place r . However, a shared resource place may have more than two holders. Thus, multiple competition paths with a same resource place as shown in Fig. 3.4 are separately used to depict the competitive relationship of every pairwise upstream and downstream activity places, and collectively used to model the common competition among all upstream and downstream activity places. Note that a competition path $t_t r t_h$ has only one upstream and only one downstream activity place with respect to r , whereas multiple competition paths with the same resource place r have multiple pairs of upstream and downstream activity places with respect to r . Therefore, a weight matrix of all competition paths

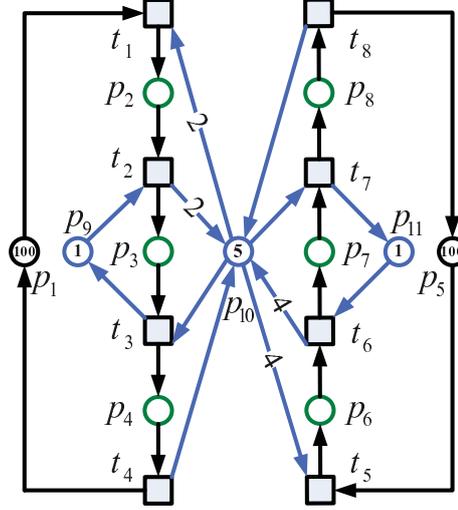


FIG. 3.6 – A WS³PR net model and two competition paths with the resource place p_{10} .

only resource places common to these two paths. Suppose that there is at least one resource, other than r_i and r_j , at which these two paths merge. This means that there exist at least two WSDCs, each containing one of r_i and r_j .

Restriction 3.1

Given a resource place r , it satisfies the following two conditions :

- (1) $(M_0(r) \bmod w_i^{in(r)}) \geq w_i^{out(r)}$, where $M_0(r)$ is the initial marking of r , and $w_i^{in(r)}$ (resp. $w_i^{out(r)}$) is the in-arc (resp. out-arc) weight of the i th competition path in L_W ; and
- (2) There exists no n -dimensional row vector $\mathbf{A} = [a_1 \dots a_i \dots a_n]$ such that $0 \leq (M_0(r) - \mathbf{A}[\mathbf{W}^{in(r)}]^T) < \min\{w_i^{out(r)}\}$, where $a_i \in \mathbb{N}$.

A competition path $t_i r t_h$ is said to satisfy Restriction 3.1 if r satisfies Restriction 3.1. Since every WSDC contains at least one competition path, a WSDC c_W is said to satisfy Restriction 3.1 if it contains a competition path $t_i r t_h$ that satisfies Restriction 3.1. It is an imposed limitation on the allocation of a resource place and its one or multiple competition paths. For example, in the above mentioned WS³PR net shown in Fig. 3.5, every WSDC contains resource place p_{13} that satisfies Restriction 3.1 at the initial marking. The other resource place p_{14} does not. Restriction 3.1 together with the concept of WSDC is adopted to enforce the liveness of WS³PR as shown in the following theorem.

3.2. INTRINSICALLY LIVE STRUCTURE

When two operations (places) are competing against each other for a commonly used resource (place), this restriction means that the rest of tokens in the resource place after an utmost use by one operation place is still enough to be used by the other operation place. It avoids the situation that a single part path (process) holds all of a kind of resources that are also required by other part paths to continue. This situation is guaranteed by the first condition of Restriction 3.1. The in-arc and out-arc of a competition path have a certain resource allocation sequence with respect to the part path(s) containing the tail or head transition. However, different in-arcs of competition paths with the same resource place do not have a certain resource allocation sequence. The second and third conditions of Restriction 3.1 make sure that at least one part path can continue after an utmost use by all holders with those in-arcs. For example, in the net model shown in Fig. 3.6(b), $M_0(p_{10}) = 5$ and $[\mathbf{W}^{(p_{10})}] = \begin{bmatrix} w_1^{in(p_{10})} & w_2^{in(p_{10})} \\ w_1^{out(p_{10})} & w_2^{out(p_{10})} \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 1 & 1 \end{bmatrix}$. It is clear that the first condition is satisfied. The second condition is also satisfied since there exists no natural number vector $A = [a_1 \ a_2]$ such that $5 = 2a_1 + 4a_2$. There are two competition paths $t_2p_{10}t_3$ and $t_6p_{10}t_7$ with the same resource place p_{10} , and the tail transition t_2 (resp. t_6) and the head one t_3 (resp. t_7) belong to the same part path, and $W(p_{10}, t_3) = 1$ (resp. $W(p_{10}, t_7) = 1$). Therefore, the resource place p_{10} satisfies Restriction 3.1.

Note that $(M_0(r) \bmod W(t_i, r)) \geq W(r, t_h)$ implies that the weight of an in-arc is greater than that of an out-arc, i.e., $W(t_i, r) > W(r, t_h)$ which is used by an algorithm to analyze the structure of WS³PR to decide the existence of a class of initial markings under which it is possible for competition paths to satisfy Restriction 3.1. For simplicity, one say that a competition path $l_{cp} = t_i r t_h$ satisfies Restriction 3.1 instead of that a competition path's resource place r satisfies Restriction 3.1.

By the following Lemmas, one can apply this restriction on competition paths found in a weighted digraph of resource places to guarantee the liveness of marked WS³PR models.

Lemma 3.2

Given a WSDC $c_W = (P_R^C, T_R^C, F_R^C, W_R^C)$, if $r_i \in P_R^C$ and $H(r_i) > 1$, then $|H_o(r_i)| = 1$.

Proof : According to the notation of $\bullet r_{io}$, $\bullet r_{io} = \{t \in T \mid \bullet t \cap P_R^C \neq \emptyset\}$. P_R^C is a simple circuit, and thus $|\bullet r_{io}| = 1$. Since every part path is a state machine, $|H(r)_{io}| = |\{p \in H(r_i) \mid p^\bullet \in \bullet r_{io}\}| = 1$.

3.2. INTRINSICALLY LIVE STRUCTURE

It indicates that every shared resource place must be used by one holder belonging to $H_o(r_i)$ and perhaps the other belonging to $H_+(r_i)$, i.e., there are at most two holders in a WSDC. Note that this lemma is of great importance in the proof of the following result.

Lemma 3.3

A simple circuit P_R^C of WSDC $c_W = (P_R^C, T_R^C, F_R^C, W_R^C)$ is never in CB if at least one competition path in c_W satisfies Restriction 3.1.

Proof : By contradiction, suppose that P_R^C is in CB and $r \in P_R^C$ is a shared resource place such that a competition path $t_t r t_h$ satisfies Restriction 3.1. According to the definition of CB and Lemma 3.2, let $H_o(r) = \{p_1\}$ and $M(p_1) \neq 0$. Hence, $M(r) = M_0(r) - kW(t_t, r) = (M_0(r) \bmod W(t_t, r)) < \min[W(r, t_h)]$ which is a contradiction.

This lemma states the relation between the net structure and initial marking for deadlock-freedom of a part path according to Lemma 2.4.

Lemma 3.4

A circular wait C is not in CB if every simple resource circuit contained in C is not in CB.

Proof : It follows from Lemmas 3.1 and 3.2.

Naturally, the following theorem reveals the relation between liveness and weights and initial marking of resource places in a marked WS³PR.

Theorem 3.2

[64] A marked WS³PR (N, M_0) is live if every WSDC contains at least one competition path that satisfies Restriction 3.1.

Proof : It follows from Theorem 2.1, and Lemmas 3.3 and 3.4.

According to Theorem 3.2, the numerical relationships among weights, and between weights and initial markings are investigated to ensure whether it is possible that every WSDC satisfies Restriction 3.1 at some particular initial markings. If so, an initial marking is then calculated through a set of developed algorithms and solving the corresponding linear programming problem. Therefore, the liveness-enforcement for a WS³PR is achieved through checking the structure of a net model, calculating and configuring a proper initial marking for resource places. Different from the siphon(monitor)-based liveness-enforcing methods, the approach further developed in

this paper unveils and takes a full advantage of the intrinsically live structure of generalized Petri nets, which is hidden behind the arc weights.

3.3 Algorithm and examples

The developed algorithms are presented and illustrated with examples in this section. Their complexity is proven to be polynomial. First, the work simplifies the given structure of a net model. It focuses on a component of resource places, their pre-sets and post-sets. It may contain the structures of circular wait that is the root of deadlock.

=====

Algorithm 3.1

Computing a weighted directed graph of resource places G_R given WS³PR N .

Input : $N = (P_0 \cup P_A \cup P_R, T, F, W)$

Output : $G_R = (P'_R, T_R, F_R, W_R)$

Step 1 : Remove all idle process places and arcs connecting to them from N ;

Step 2 : Remove all operation places and arcs connecting to them from N ;

Step 3 : Remove all transitions without preset or postset and arcs connecting to these transitions from N ;

Step 4 : Remove all resource places without input or output transitions and arcs connecting to these places ;

Step 5 : G_R is the remaining part of N .

=====

The work in thesis uses $\text{Compute_GR}(N)$ to denote a function that implements the above algorithm. For the structure of WS³PR shown in Fig. 3.7(a), its weighted digraph of resource places G_R shown in Fig. 3.7(b) is obtained by running this function. Its complexity is $O(n)$, where $n = |P| + |T|$ is the size of a given WS³PR. Second, all resource paths should be found from G_R by

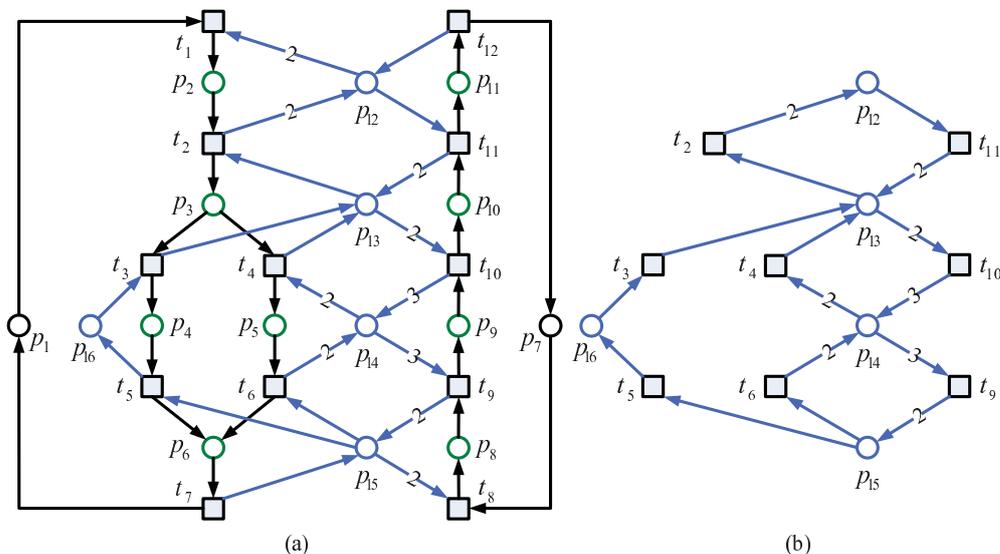


FIG. 3.7 – (a) A WS³PR net model N with 5 resource places and (b) G_R derived from N .

checking their weights of in-arcs and out-arcs.

=====

Algorithm 3.2

Computing all resource paths from a weighted digraph of resource places G_R .

Input : $G_R = (P'_R, T_R, F_R, W_R)$

Output : L

$L := \emptyset$; $Tail := \emptyset$; $T_{imp} := \emptyset$;

while $(T_R - Tail \neq \emptyset)$

{

$T_{imp} := \emptyset$;

choose a t_i from $T_R - Tail$;

while $(T_R - \{t_i\} - T_{imp} \neq \emptyset)$

{

choose a t_h from $T_R - \{t_i\} - T_{imp}$;

if $(t_i^\bullet \cap {}^\bullet t_h = \{r\})$

```

        {  $\mathbf{L} := \mathbf{L} \cup \{t_t, r, t_h\}; T_{tmp} := T_{tmp} \cup \{t_h\};$ 
      }
      else
         $T_{tmp} := T_{tmp} \cup \{t_h\}$ 
      }
       $Tail := Tail \cup \{t_t\};$ 
    }

  Output  $\mathbf{L}$ 

```

=====

In the above algorithm, T_R is the set of all transitions in G_R , and $Tail$ means a set of tail transitions, and T_{tmp} is also a set of transitions that is temporally used to store tested transitions. We use $C_All_RP(G_R)$ to denote a function that implements this algorithm. Since every resource path begins and ends with a tail and head transition, the number of all resource paths is $|T_R|^2 - |T_R|$ at the worst case. Hence, this algorithm's complexity is $O(n^2)$. Given G_R in Fig. 3.7(b), $\mathbf{L} = C_All_RP(G_R) = \{t_2p_{12}t_{11}; t_{11}p_{13}t_2; t_3p_{13}t_{10}; t_4p_{13}t_{10}; t_3p_{13}t_2; t_4p_{13}t_2; t_{11}p_{13}t_{10}; t_5p_{16}t_3; t_{10}p_{14}t_4; t_6p_{14}t_9; t_6p_{14}t_4; t_{10}p_{14}t_9; t_9p_{15}t_5; t_9p_{15}t_6\}$.

Third, the outcome of $Get_All_RP(G_R)$ is partitioned into two sets, \mathbf{L}_W and \mathbf{L}_F . All elements in \mathbf{L}_W are competition paths whose in-arcs have greater weights than those of out-arcs, and those in \mathbf{L}_F are the rest of resource paths in \mathbf{L} . It is possible for the resource paths in \mathbf{L}_W only to find a class of initial markings satisfying Restriction 3.1.

=====

Algorithm 3.3

Partition set \mathbf{L} into \mathbf{L}_W in which every resource path has a greater weight of in-arc than that of out-arc, and $\mathbf{L}_F = \mathbf{L} - \mathbf{L}_W$.

Input : \mathbf{L}

Output : \mathbf{L}_W and \mathbf{L}_F

while ($\mathbf{L} \neq \emptyset$)

3.3. ALGORITHM AND EXAMPLES

```

{
  choose a resource path  $l = t_t r t_h$  from  $\mathbf{L}$ ;
  if ( $W(t_t, r) > W(r, t_h)$ )
    { $\mathbf{L}_W := \mathbf{L}_W \cup \{l\}$ ;  $\mathbf{L} := \mathbf{L} - \{l\}$ ;}
  else
    { $\mathbf{L}_F := \mathbf{L}_F \cup \{l\}$ ;  $\mathbf{L} := \mathbf{L} - \{l\}$ ;}
}

```

Output \mathbf{L}_W and \mathbf{L}_F ;

=====

Let $\text{Partition_RP}(\mathbf{L})$ denote a function implementing this algorithm. Its complexity is also $O(n)$. For example, \mathbf{L} in Fig. 3.7(b) is partitioned into $\mathbf{L}_F = \{t_3 p_{13} t_{10}; t_4 p_{13} t_{10}; t_3 p_{13} t_2; t_4 p_{13} t_2; t_{11} p_{13} t_{10}; t_5 p_{16} t_3; t_6 p_{14} t_9; t_6 p_{14} t_4; t_{10} p_{14} t_9\}$ and $\mathbf{L}_W = \{t_2 p_{12} t_{11}; t_{11} p_{13} t_2; t_{10} p_{14} t_4; t_9 p_{15} t_5; t_9 p_{15} t_6\}$.

=====

Algorithm 3.4

Analyzing the structure with respect to every resource place r belonging to competition paths in \mathbf{L}_W to decide which competition paths can never satisfy Restriction 3.1.

Input : \mathbf{L}_W

Output : \mathbf{L}'_W

$\mathbf{L}'_W = \emptyset$; $Res := \text{Get_Resource}(\mathbf{L}_W)$;

for ($k = 1 ; k \leq |Res| ; k++$)

{

$M_0(r_k) := 0$;

if (r_k does not satisfy the 3rd condition of Restriction 3.1)

$\mathbf{L}'_W = \mathbf{L}'_W \cup \text{Get_All_CP}(r_k)$;

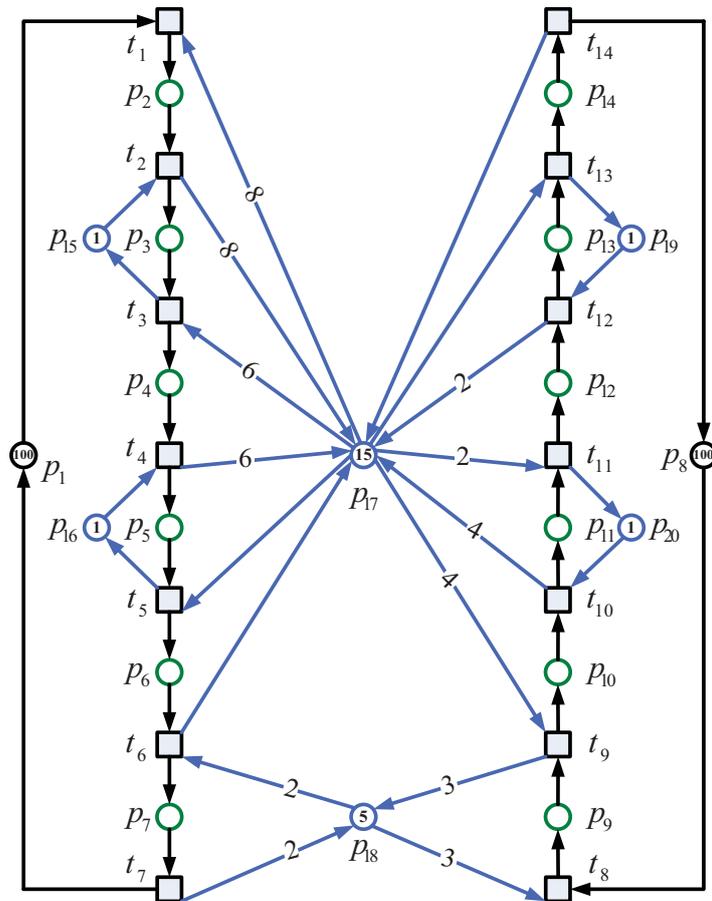


FIG. 3.8 – There exists a proper initial marking of p_{17} to satisfy Restriction 3.1.

3.3. ALGORITHM AND EXAMPLES

else

{

$$U = 2(w_1^{in(r_k)} \times \dots \times w_i^{in(r_k)} \times \dots \times w_n^{in(r_k)});$$

$$c = U/w_n^{in(r_k)};$$

for ($j_1 = 1; j_1 \leq c; j_1 ++$)

{

for ($rem = w^{out(r_k)_n}; rem \leq (w^{in(r_k)_n} - 1); rem ++$)

{

$Fail := 0;$

for ($j_2 = 1; j_2 \leq n - 1; j_2 ++$)

{

if ($((j_1 \times w_n^{in(r_k)} + rem) < w_{j_2}^{out(r_k)})$)

$Fail := 1; \mathbf{break};$

}

if ($Fail == 1$ and $(j_1 \times w_n^{in(r_k)} + rem)$ satisfies the second condition of Restriction 3.1)

$M_0(r_k) := j_1 \times w_n^{in(r_k)} + rem; \mathbf{break};$

}

if ($M_0(r_k) > 0$)

break;

}

if ($M_0(r_k) == 0$)

$L'_W = L'_W \cup Get_All_CP(r_k);$

}

}

Output L'_W .

3.3. ALGORITHM AND EXAMPLES

=====

Analyze_LW(\mathbf{L}_W) is used to denote a function that implements the above algorithm. $Get_Resource(\mathbf{L}_W)$ is a function that obtains a set Res of all resource places used by competition paths in \mathbf{L}_W , $Get_All_CP(r_k)$ is a function that obtains all competition paths in \mathbf{L}_W with resource place r_k , and U is the upper limit of the computation. In the example shown in Fig. 3.8, $[\mathbf{W}^{(p_{17})}] = \begin{bmatrix} 2 & 4 & 6 & 8 \\ 1 & 2 & 1 & 6 \end{bmatrix}$, and $M_0(p_{17}) = 15$ is computed according to Algorithm 3.4. Apparently, $M_0(p_{17}) = 15$ makes all competition paths with p_{17} , i.e., $t_{12}p_{17}t_{13}$, $t_{10}p_{17}t_{11}$, $t_4p_{17}t_5$, and $t_2p_{12}t_3$, satisfying Restriction 3.1. This algorithm's complexity is $O(n^4)$.

Next, an algorithm checking set $\mathbf{L}_F^* = \mathbf{L}_F \cup \mathbf{L}'_W$ is presented to decide whether there exists a class of initial markings given a WS³PR under which it is live. In the algorithm, $Head$ and $Tail$ are sets of head and tail transitions obtained via functions $C_Head(\mathbf{L}_F^*)$ and $C_Tail(\mathbf{L}_F^*)$, respectively. For example, $\mathbf{L}_F^* = \{t_3p_{13}t_{10}; t_4p_{13}t_{10}; t_3p_{13}t_2; t_4p_{13}t_2; t_{11}p_{13}t_{10}; t_5p_{16}t_3; t_6p_{14}t_9; t_6p_{14}t_4; t_{10}p_{14}t_9\}$, $C_Head(\mathbf{L}_F^*) = \{t_3, t_4, t_5, t_6, t_{10}, t_{11}\}$ and $C_Tail(\mathbf{L}_F^*) = \{t_2, t_3, t_4, t_9, t_{10}\}$.

=====

Algorithm 3.5

Analyzing set \mathbf{L}_F^* to see whether there is a proper initial marking under which a WS³PR is live.

Input : $\mathbf{L}_F^* = \mathbf{L}_F \cup \mathbf{L}'_W$

Output : Yes/No

while ($\mathbf{L}_F^* \neq \emptyset$)

{

$Tail := Get_Tail(\mathbf{L}_F^*);$

$Head := Get_Head(\mathbf{L}_F^*);$

if ($Tail - Head = \emptyset$ and $Head - Tail = \emptyset$)

 Output : No ;

else

 { **if** ($Tail - Head \neq \emptyset$)

3.3. ALGORITHM AND EXAMPLES

```

    remove all resource paths whose tail transitions are in  $Tail - Head$  from  $\mathbf{L}_F^*$ ;
    if ( $Head - Tail \neq \emptyset$ )
        remove all resource paths whose head transitions are in  $Head - Tail$  from  $\mathbf{L}_F^*$ ;
    }

```

Output : Yes.

=====

The function $Analyze_LF(\mathbf{L}_F^*)$ implementing Algorithm 3.5 analyzes \mathbf{L}_F^* and decides whether they can form a strongly connected component that must contain at least one WSDC, in which no competition path's resource place can satisfy Restriction 3.1. If there is no strongly connected component formed by resource paths in \mathbf{L}_F^* , it implies that one can find an initial marking to make the resource place of at least one competition path in every WSDC satisfy Restriction 3.1. Hence, the marked WS^3PR is live. Apparently, this algorithm's complexity is $O(n)$.

Once the result of Algorithm 3.5 is "Yes", one can give a proper initial marking immediately. For $r_k \in Res$, where Res is the output of the function $Get_Resource(\mathbf{L}_W)$, $M_0(r_k)$ has been computed by Algorithm 3.4, and the others can be any positive integer.

Finally, one combines all algorithms to show a complete application of the proposed method in Algorithm 3.6 that can be finished in polynomial time.

=====

Algorithm 3.6

Checking the structure of a WS^3PR to decide whether there exist initial markings under which WS^3PR is live.

Input : N

Output : M_0

Step 1 : $G_R := Compute_GR(N)$;

Step 2 : $\mathbf{L} := C_All_RP(G_R)$;

Step 3 : $\{\mathbf{L}_F, \mathbf{L}_W\} := Partition_RP(\mathbf{L})$;

3.3. ALGORITHM AND EXAMPLES

Step 4 : Analyze_LW(\mathbf{L}_W);

Step 5 : Analyze_LF(\mathbf{L}_F^*);

Step 6 : **If** there exist such a kind of initial markings, give the minimal one, i.e., for $r_k \in Res$, $M_0(r_k)$ has been computed by Analyze_LW(\mathbf{L}_W), and the others can be any positive integer.

=====

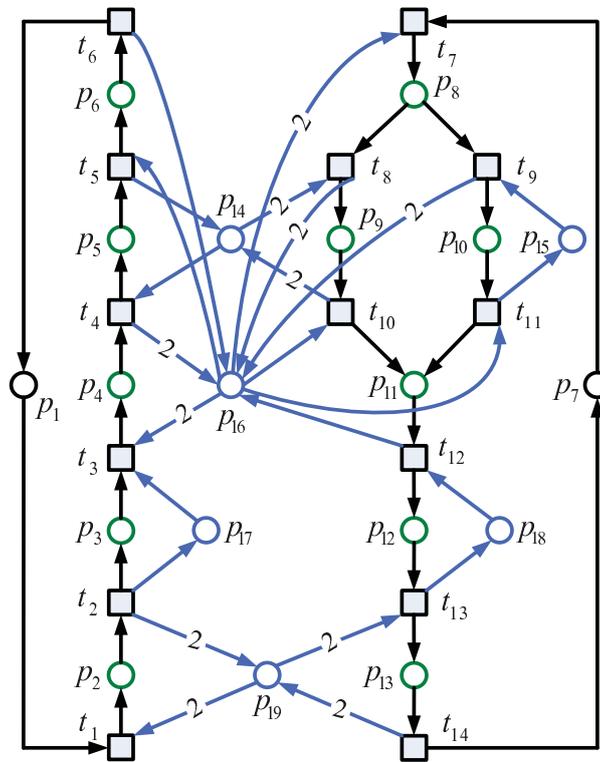


FIG. 3.9 – A WS³PR with six resource places.

Apply this method to check the structure of WS³PR in Fig. 3.9. The resource paths in \mathbf{L}_F^* form a strongly connected component, and also a WSDC in this example, $t_3p_{17}t_2p_{19}t_{13}p_{18}t_{12}p_{16}t_3$. No resource place of its competition paths can satisfy Restriction 3.1, i.e., regardless initial markings, this WSDC's competition path's resource place can do so. In contrast with the existing structure, if the weights of arcs (p_{19}, t_{13}) and (t_{14}, p_{19}) changed to one, the result is different. In this case, the competition path $t_{13}p_{19}t_2$ may satisfy Restriction 3.1 with a proper initial marking, and furthermore

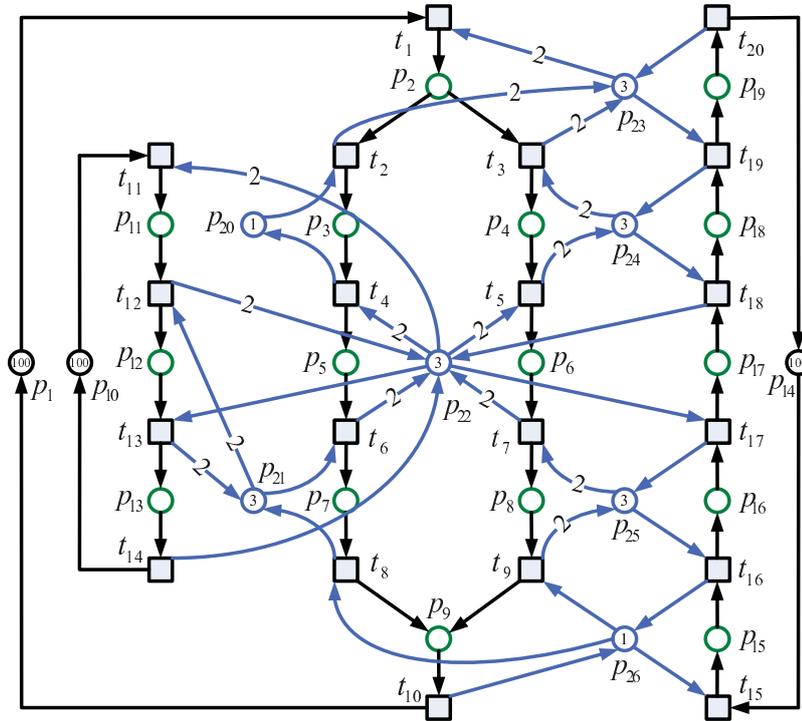


FIG. 3.10 – A WS^3PR model with 9 WSDCs in its G_R .

this new marked WS^3PR is live.

The WS^3PR in Fig. 3.10 is another example from which nine WSDCs can be found in its weighted digraph of resource places. The WSDCs and their respective competition paths that satisfy Restriction 3.1 are enumerated in Table 3.1. M_0 for resource places $p_{21} - p_{25}$ obtained are shown in Table 3.2. The initial marking for $p_1, p_{10}, p_{17}, p_{20}$ and p_{26} can be any positive integer. Hence, the net model in Fig. 3.10 is live under the initial marking $M_0 = 100p_1 + 100p_{10} + 100p_{17} + p_{20} + 3p_{21} + 3p_{22} + 3p_{23} + 3p_{24} + 3p_{25} + p_{26}$.

3.4 Discussion

Compared with the structure of ordinary Petri nets, the role of weights of arcs in determining the liveness of generalized ones should not be neglected. Unlike all arcs' weights being one in

3.4. DISCUSSION

TAB. 3.1 – WSDCs and their respective competition paths that satisfy Restriction 3.1 in the WS³PR of Fig. 3.10

No.	WSDC	l_{CP} in \mathbf{L}_W
1	$t_{13}p_{21}t_{12}p_{22}t_{13}$	$t_{12}p_{22}t_{13}$
2	$t_{13}p_{21}t_6p_{22}t_{13}$	$t_{13}p_{21}t_6$
3	$t_7p_{22}t_{17}p_{25}t_7$	$t_7p_{22}t_{17}$
4	$t_{18}p_{22}t_5p_{24}t_{18}$	$t_5p_{24}t_{18}$
5	$t_3p_{23}t_{19}p_{24}t_3$	$t_3p_{23}t_{19}$
6	$t_9p_{25}t_{16}p_{26}t_9$	$t_9p_{25}t_{16}$
7	$t_4p_{20}t_2p_{23}t_{19}p_{24}t_{18}p_{22}t_4$	$t_2p_{23}t_{19}$
8	$t_8p_{21}t_{12}p_{22}t_{17}p_{25}t_{16}p_{26}t_8$	$t_{12}p_{22}t_{17}$
9	$t_8p_{21}t_6p_{22}t_{17}p_{25}t_{16}p_{26}t_8$	$t_6p_{22}t_{17}$

TAB. 3.2 – The minimal $M_0(r_i)$ for resource places used by competition paths in Tabel 3.1

r_i	competition path	$M_0(r_i)$
p_{21}	$t_{13}p_{21}t_6$	3
p_{22}	$t_{12}p_{22}t_{13}$, $t_7p_{22}t_{17}$, $t_{12}p_{22}t_{17}$, and $t_6p_{22}t_{17}$	3
p_{23}	$t_3p_{23}t_{19}$ and $t_2p_{23}t_{19}$	3
p_{24}	$t_5p_{24}t_{18}$	3
p_{25}	$t_9p_{25}t_{16}$	3

ordinary S³PR, weights of arcs in WS³PR mean the multiple use of resources by an operation, and the numerical relationship between weights of arcs connecting operation and resource places, and token counts in resource places restrict the allocation of system resources. Inspired by concepts of circular wait and circular blocking presented in [52], the chapter proposes the concept of weighted simple directed circuit (WSDC) that is a generalized structure of circular wait. Different from the circular wait that is a strongly connected component and may be composed of several non-disjoint simple resource circuits [52], WSDC is a simple resource circuit. Thus a shared resource place may be competed by only two operations in such a circuit. It is important for us to study the numerical relationship between weights and initial marking. Hence, a new deadlock prevention method may be developed from the view point of WSDC rather than siphons. The effect of weights is also considered in defining the state of circular block for WSDC in this research. It is similar to the definition of insufficient marking for a siphon in a generalized net [91]. A generalized version of Dining Philosophers modeled by WS³PR is shown in Fig. 3.11. $c_W = t_2p_{16}t_{14}p_{20}t_{11}p_{19}t_8p_{18}t_5p_{17}t_2$ is the only WSDC in this net. When it is in the state of CB under the current initial marking

3.4. DISCUSSION

$M_0 = 30p_1 + 30p_4 + 30p_7 + 30p_{10} + 30p_{13} + 2r_1 + 1r_2 + 2r_3 + 7r_4 + 4r_5$, all five part paths are dead, i.e., the entire system is deadlocked. In this case, there is one token trapped in p_{19} since the minimal weight of out-arcs is greater than the number of trapped tokens. Evidently, this undesirable state is a result of combined action of its net structure, weights, and initial marking.

After the seminal work of Ezpeleta [16], deadlock prevention methods are implemented by adding monitors (control places) and corresponding arcs to original model plants according to respective policies to enforce liveness [58]. The structural complexity of the controlled system becomes an important issue for the policies developed in the literature. The key difference between the proposed method and the existing ones is that the deadlock prevention is implemented through configuring a proper initial marking for WS³PR with certain characteristic structure, i.e., at least one competition path has an in-arc weighting over its corresponding out-arc in every WSDC. The method cannot replace others in all situations but can be meaningful before other policies are applied. This method is different from the pioneered work of Zhou *et al.* [124] and [125] in which an initial marking is found to ensure the net's properties given a special net structure and quantity of shared resources. The kernel of our work is to identify the numerical relationship between weights representing multiple resource requirements and initial marking of resource places to decide whether certain restriction are met. For example, before applying any deadlock prevention method under the given initial marking, the structure should be examined to decide whether it is live with certain initial markings. Obviously, two competition paths, both $l_{CP1} = t_{14}p_{20}t_{11}$ and $l_{CP2} = t_{11}p_{19}t_8$ have in-arcs weighting over their corresponding out-arcs in the unique WSDC, $c_W = t_2p_{16}t_{14}p_{20}t_{11}p_{19}t_8p_{18}t_5p_{17}t_2$. One can make it live by re-setting an initial marking as $M_0 = 30p_1 + 30p_4 + 30p_7 + 30p_{10} + 30p_{13} + 2r_1 + 1r_2 + 2r_3 + 5r_4 + 3r_5$. Compared with the original one $M_0(P_R) = 2r_1 + 1r_2 + 2r_3 + 7r_4 + 4r_5$, the new one decreases some resources' capacity (i.e., $M_0(P_R) = 2r_1 + 1r_2 + 2r_3 + 5r_4 + 3r_5$). Most importantly, the new one makes the net live. This can save hardware and software cost caused by the monitors and arcs when siphon-based deadlock prevention methods are used. As another example, the net model in Fig. 3.10 has 18 strict minimal siphons and 6 elementary siphons. Hence if a siphon-based method is used, a significant number of monitors and arcs are needed. Also most of existing methods are of exponential complexity [23], [58], and [98], and only few are polynomial [78].

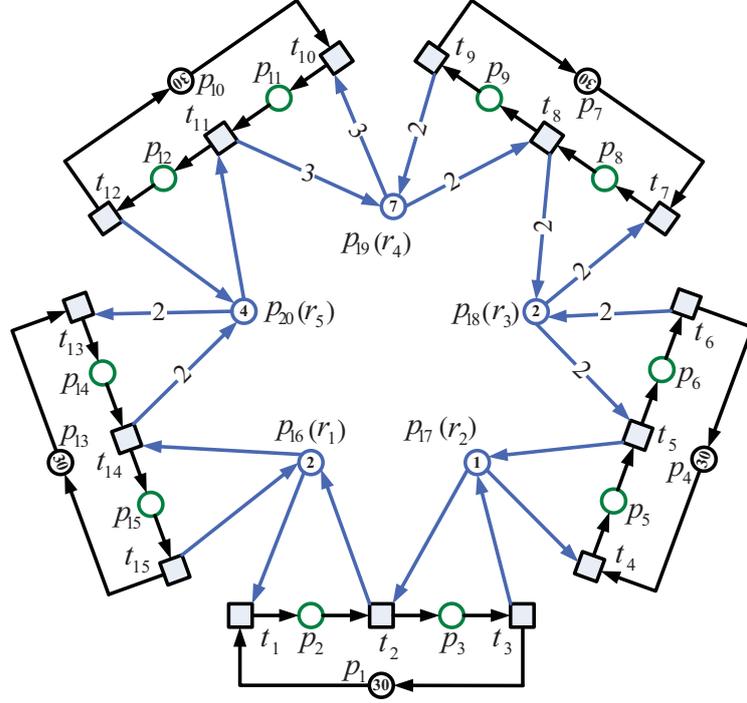


FIG. 3.11 – A WS³PR modeling the problem of extended Dining Philosophers.

Actually, according to Theorem 3.2, a marked WS³PR is deadlock-free and live if only one competition path satisfies Restriction 3.1 rather than all those do in each single WSDC. For example, both $l_{CP1} = t_{14}p_{20}t_{11}$ and $l_{CP2} = t_{11}p_{19}t_8$ whose in-arcs weigh over out-arcs, may satisfy Restriction 3.1 with respect to a proper initial marking in WSDC $c_W = t_2p_{16}t_{14}p_{20}t_{11}p_{19}t_8p_{18}t_5p_{17}t_2$ of Fig. 3.11. Either choice will be enough to make the resulting model live. In other words, $M'_0 = 30p_1 + 30p_4 + 30p_7 + 30p_{10} + 30p_{13} + 2r_1 + 1r_2 + 2r_3 + 3r_4 + 3r_5$ and $M''_0 = 30p_1 + 30p_4 + 30p_7 + 30p_{10} + 30p_{13} + 2r_1 + 1r_2 + 2r_3 + 5r_4 + 2r_5$ are both such proper initial markings. This can reduce the system's dependence on resources. However, no matter only one or more competition paths in every WSDC satisfy Restriction 3.1, the number of initial markings of idle process places can be any positive integer. Some tokens will never enter the processing system if $M_0(p_{0_i})$ is too large, i.e., $M(p_{0_i}) > 0, \forall M \in R(N, M_0)$. In fact, the maximal number of job instances that can enter a system is equal to the sum of all operation places' bounds.

To avoid the enumeration of all simple circuits (WSDCs) in a weighted digraph of resource places, which has exponential complexity at the worst case, one can set an initial marking to

3.5. SUMMARY

make every competition path in $\mathbf{L}_W - \mathbf{L}'_W$ satisfy Restriction 3.1. Therefore, determining how to refine competition paths in $\mathbf{L}_W - \mathbf{L}'_W$ without obtaining all simple circuits in prior is an interesting problem and still open.

Finally, this method can be also employed in the stage of model design. A structure of WS³PR is designed according to some specifications describing a practical system. If permitted, the structure and weights of arcs should be properly designed to build an inherently deadlock-free model, thereby saving control cost.

3.5 Summary

A new method is provided for deadlock prevention that uses the concepts of generalized circular wait and circular blocking. In WS³PR, it is proved that a net model with a certain characteristic structure is deadlock-free and live with respect to a class of initial markings. A set of polynomial algorithms are developed to implement this method. They can be easily used before any attempt to use other deadlock prevention policies. As a result, one may significantly save the deadlock controller design efforts and implementation cost.

Chapitre 4

Hybrid Liveness-enforcing Method for WS^3PR

This chapter proposes a hybrid liveness-enforcing method for WS^3PR , which can well model many flexible manufacturing systems. The proposed method combines elementary siphons with a characteristic structure-based method to prevent deadlocks and enforce liveness to the net class under consideration. The characteristic structure-based method addressed in the previous chapter is further advanced in this chapter. It unveils and takes a full advantage of an intrinsically live structure (ILS) of generalized Petri nets, which is hidden behind the arc weights, to achieve the liveness enforcement without any external control agent such as monitors. This hybrid method can identify and remove redundant monitors from a liveness-enforcing supervisor designed according to existing policies, improve the permissiveness, reduce the structural complexity of a controlled system, and consequently save the control implementation cost.

4.1 Introduction

The previous chapter unveils and takes a full advantage of an intrinsically live structure of WS^3PR , which is hidden behind the arc weights, to achieve liveness enforcement without external monitors. Instead of siphons, WSDCs are a new kind of structural objects and adopted to express augmented circular waits of resources. The liveness-enforcement of a WS^3PR with an intrinsically

live structure is realized by reconfiguring the number of resources to make all WSDCs satisfy a proposed restriction. The work in Chapter 3 is devoted to liveness-enforcement of WS^3PR by purely applying the WSDC-based method. In this chapter, the ILS/WSDC-based method is developed. It is found that the identification of an intrinsically live structure and calculation of proper initial number of tokens can be attributed to a typical congruence problem in number theory. A mathematical programming is designed to solve the problem in the chapter. Since the pure application of the WSDC-based method demands that all WSDCs in a WS^3PR should first satisfy some structural conditions, which greatly limits its application scope, a hybrid liveness-enforcing method is proposed by combining the WSDC-based one with the well-accepted elementary siphons-based one [54] [58] [59]. The hybrid one no longer demands a global intrinsically live structure of WS^3PR , and therefore, the application of WSDCs is extended. Theorems proposed in this chapter build a bridge between WSDCs and siphons to realize this hybrid method. In addition, the minimally marked and controlled siphons are analyzed with the assistance of WSDCs. The hybrid method can simplify the liveness-enforcing supervisor for a system, improve the permissiveness, and finally, reduce the control implementation cost.

4.2 Liveness-enforcement in WS^3PR

This section first briefly goes over the liveness-enforcing method based on elementary siphons [53] [54] [61] to make the chapter self-contained and then elaborates on and further develops a characteristic structures-based liveness-enforcing method along the lines of Chapter 3.

4.2.1 Liveness-enforcement based on elementary siphons

Both theory of regions and structural theory are the frequently used techniques in deadlock control and liveness-enforcement of Petri nets. It is undeniable that the former that grounds upon a reachability graph is more reliable, effective, and accurate [23] [98]. It achieves its goal by exploring a complete state space. However, it has a fatal flaw. Since the number of states of a system usually grows exponentially with the system size, its use becomes impossible for a large-scale or even a moderate model. The state explosion problem is a conundrum that currently seems to have no efficient solution. The past two decades witnessed the widespread adoption of the structural

theory as a tool in deadlock prevention. It works by fully utilizing the structural information of a Petri net without the help of its reachability graph.

As one of the most important concepts of the structural theory, siphons become the cornerstone of various deadlock prevention policies [11] [24], [37], [52], and [60]. In [16], the deadlock prevention is implemented by separately adding a monitor for every strict minimal siphon. As gradually recognized, this method suffers from a number of problems, one of which is the supervisor's structural complexity. It may even lead to a much more structurally complex supervisor than the original plant net model. Aiming at solving this problem, Li and Zhou pioneered in elementary siphons [53] [54] [58] [59] [61]. Through fully considering the structural relationship among all strict minimal siphons, they divide them into two classes, i.e., elementary and dependent ones. The latter can be obtained by linear combinations of the former and can be further classified into being strongly and weakly dependent with respect to the former. It has been shown that the number of elementary siphons in a net model is bounded by the smaller of place and transition counts. Not all but only elementary siphons need to be explicitly supervised by the corresponding monitors to enforce the liveness. The controllability of dependent siphons can be ensured by properly setting the initial numbers of tokens in the monitors for elementary ones. In other words, the dependent siphons can be implicitly controlled.

For example, the WS³PR shown in Fig. 4.1(a) has the following place partition : $P_0 = \{p_1, p_7\}$, $P_A = \{p_2 - p_6, p_8 - p_{12}\}$, and $P_R = \{p_{13} - p_{17}\}$. There are totally 36661 reachable states and 35 dead ones among them.

There are 10 strict minimal siphons as listed below. By the definition of elementary siphons in [54] [53], S_1 - S_4 can be chosen to be elementary, while S_5 - S_{10} are strongly dependent and marked with *.

$$S_1 = \{p_6, p_9, p_{16}, p_{17}\},$$

$$S_2 = \{p_5, p_{10}, p_{15}, p_{16}\},$$

$$S_3 = \{p_4, p_{11}, p_{14}, p_{15}\},$$

$$S_4 = \{p_3, p_{12}, p_{13}, p_{14}\},$$

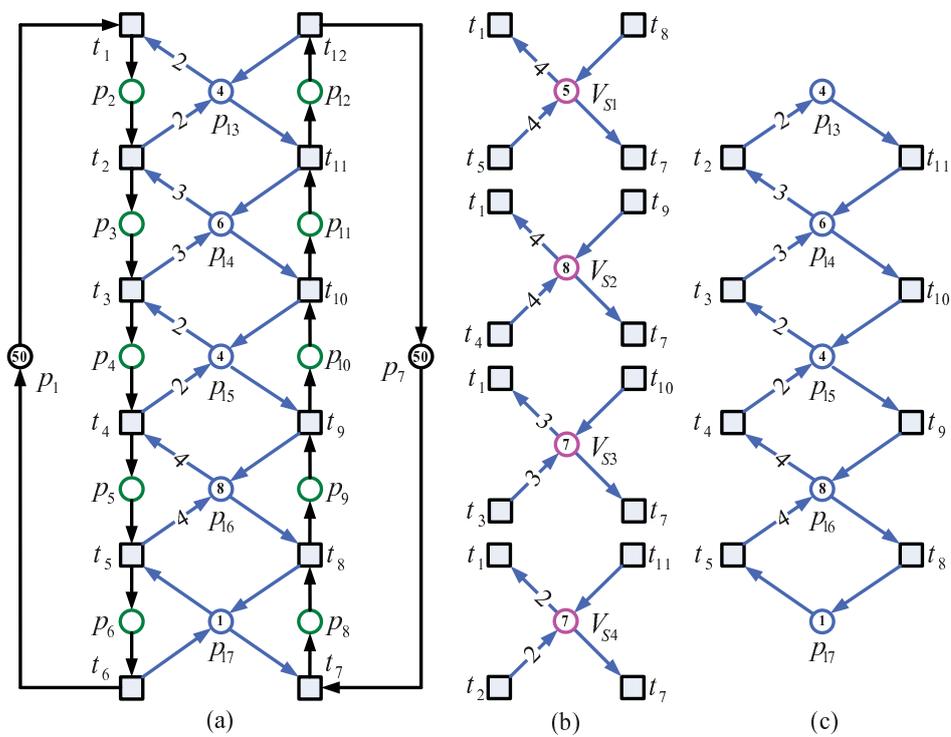


FIG. 4.1 – (a) A WS³PR with five resource places, (b) four monitors designed according to elementary siphons, and (c) the corresponding weighted digraph of resource places.

$$S_5^* = \{p_6, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}, p_{17}\},$$

$$S_6^* = \{p_6, p_{11}, p_{14}, p_{15}, p_{16}, p_{17}\},$$

$$S_7^* = \{p_6, p_{10}, p_{15}, p_{16}, p_{17}\},$$

$$S_8^* = \{p_5, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}\},$$

$$S_9^* = \{p_5, p_{11}, p_{14}, p_{15}, p_{16}\}, \text{ and}$$

$$S_{10}^* = \{p_4, p_{12}, p_{13}, p_{14}, p_{15}\}.$$

We add monitors for all four elementary ones, as shown in Fig. 4.1(b), where

$$\bar{S}_1 = 4p_5 + p_8, M(V_{S1}) = 9 - 4 = 5;$$

$$\bar{S}_2 = 2p_4 + p_9, M(V_{S2}) = 12 - 4 = 8;$$

$$\bar{S}_3 = 3p_3 + p_{10}, M(V_{S3}) = 10 - 3 = 7; \text{ and}$$

$$\bar{S}_4 = 2p_2 + p_{11}, M(V_{S4}) = 10 - 3 = 7.$$

This controlled system is live but has 4421 states only. Obviously, the current liveness-enforcing method based on elementary siphons greatly restricts the system's behavior of WS³PR. One of the reasons is the existence of redundant monitors.

4.2.2 Liveness-enforcement based on intrinsically live structures

Being totally different from the siphon-monitor-based liveness-enforcing methods in the literature, the method in Chapter 3 is proposed by considering both structure and dynamic properties given a net model. According to the theory, a net model with a characteristic structure is live at some particular initial markings. Applying this method to a WS³PR model, its weighted digraph of resource places is firstly analyzed. Instead of siphons, the concept of WSDCs that are augmented and generalized simple CWs is adopted to identify this class of characteristic structures. More explicitly, the competitive relationship between the upstream and downstream activity places of a competition path is thoroughly investigated. Given a competition path $l_{cp} = t_t r t_h$, the in-arc (t_t, r) and out-arc (r, t_h) depict the upstream and downstream operations collectively using the same resource place r , respectively. Moreover, $W(t_t, r)$ ($W(r, t_h)$) represents the required number of tokens

for completing every single upstream (downstream) operation.

For example, in the WSDC shown in Fig. 4.2(c), $l_{cp} = t_2 p_6 t_3$ is the only competition path. The weight of the in-arc (t_2, p_6) (resp. out-arc (p_6, t_3)) represents that the upstream (resp. downstream) activity place p_2 (p_4) needs three (resp. two) tokens in the resource place p_6 to complete an operation. Once the tokens in r are all requested and occupied by the upstream activity place or the remaining tokens in r are not enough to be used by the downstream activity place at least once, a CB and potential deadlock may occur. Therefore, the numerical relation among in-arc and out-arc weights and initial token counts in the resource place turns out to be useful to control resource allocation without any external monitor, which cannot be realized in ordinary Petri nets. Thus, a restrictive condition on both necessary characteristic structure and proper initial marking is laid down to ensure that the upstream activity place cannot overly require and occupy tokens in the resource place. In other words, even if the upstream activity place is allowed to take priority over the downstream one to maximally require and occupy the tokens in the resource place, the remaining ones that cannot be obtained by the upstream activity place due to a potential barrier formed by the weight of in-arc, are still adequate to complete at least once the downstream operation.

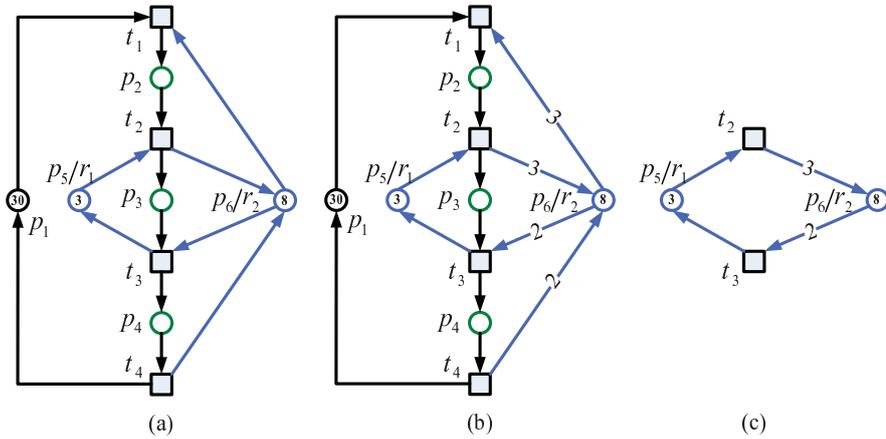


FIG. 4.2 – (a) An ordinary net, (b) a generalized net, and (c) a weighted digraph of resource places, G_R .

Naturally, this restrictive condition is abstracted to a modular arithmetic, i.e., $M_0(r) \bmod W(t_i, r) \geq W(r, t_h)$. Here, \bmod is the modulus operator that gives the remainder after an integer division. Let a be an integer and b a positive integer greater than 1. We define $a \bmod b$ to be the remainder c

when a is divided by b . That is to say $c = a \bmod b = a - b\lfloor a/b \rfloor$, where $\lfloor a/b \rfloor$ means the largest integer less than or equal to a/b , for example, $\lfloor 1/2 \rfloor = 0$ and $\lfloor 8/3 \rfloor = 2$. The condition implies a two-fold limitation on both structure and initial marking. First, the in-arc weight must be larger than the out-arc one, which forms the so-called potential barrier, i.e., a class of characteristic and intrinsically live structures. Second, the in-arc weight dose not divide the initial marking and the remainder must be larger or equal to the out-arc weight. Take the aforementioned competition path $l_{cp} = t_2p_6t_3$ as an example. If the initial number of tokens in p_6 is nine, all of them may be occupied by the upstream activity place p_2 of l_{cp} . However, if p_6 is configured with eight tokens at the initial marking, it guarantees that the downstream activity place p_4 can obtain at least two tokens to complete an operation.

Let a and c be integers and b a positive integer. We say that a is congruent to c modulo b , denoted by $a \equiv c \pmod{b}$, if $a \equiv c \pmod{b} \Leftrightarrow a = kb + c$, where $k = \lfloor a/b \rfloor$, and $a/b - 1 < k \leq a/b$. For more details about theory of congruences, please see [115].

Theorem 4.1

[115] If b_1, b_2, \dots , and b_n are pairwise relatively prime and greater than 1, and c_1, c_2, \dots , and c_n are any integers, then there is a solution a to the following simultaneous congruences :

$$\begin{cases} a \equiv c_1 \pmod{b_1} \\ a \equiv c_2 \pmod{b_2} \\ \dots \\ a \equiv c_n \pmod{b_n} \end{cases} \quad (4.1)$$

If a and a' are two solutions, then $a \equiv a' \pmod{b}$, where $b = \prod_{i=1}^n b_i$.

Hence, a proper $M_0(r)$ such that every column of the weight matrix satisfies the above restrictive conditions at the same time is a solution of the above simultaneous congruences.

Remark 4.1

[115] If the system of the linear congruences (1) is soluble, then its solution can be conveniently described as follows :

$$a \equiv \sum_{i=1}^n c_i B_i B'_i \pmod{b} \quad (2)$$

4.2. LIVENESS-ENFORCEMENT IN WS³PR

where $b = \prod_{i=1}^n b_i$, $B_i = b/b_i$, and $B'_i = B_i^{-1} \pmod{b_i}$, $i \in \{1, 2, \dots, n\}$.

This result gives a general solution of Theorem 4.1. For example, as for the weight matrix $[\mathbf{W}^{(r)}] = \begin{bmatrix} 5 & 7 & 8 \\ 3 & 2 & 1 \end{bmatrix}$, a proper $M_0(r)$ is a solution of the following simultaneous congruences.

$$\begin{cases} M_0(r) \equiv 3 \pmod{5} \\ M_0(r) \equiv 2 \pmod{7} \\ M_0(r) \equiv 1 \pmod{8} \end{cases}$$

By Eq. (2), we have

$$\begin{aligned} w &= \prod_{i=1}^n w_i^{in(r)} = 5 \cdot 7 \cdot 8 = 280, \\ W_1 &= w/w_1^{in(r)} = 280/5 = 56, \\ W_2 &= w/w_2^{in(r)} = 280/7 = 40, \\ W_3 &= w/w_3^{in(r)} = 280/8 = 35, \\ W'_1 &= W_1^{-1} \pmod{w_1^{in(r)}} = 56^{-1} \pmod{5} = 1, \\ W'_2 &= W_2^{-1} \pmod{w_2^{in(r)}} = 40^{-1} \pmod{7} = 3, \\ W'_3 &= W_3^{-1} \pmod{w_3^{in(r)}} = 35^{-1} \pmod{8} = 3. \end{aligned}$$

Hence,

$$\begin{aligned} M_0(r) &= w_1^{out(r)} W_1 W'_1 + w_2^{out(r)} W_2 W'_2 \\ &\quad + w_3^{out(r)} W_3 W'_3 \pmod{w} \\ &= 3 \cdot 56 \cdot 1 + 2 \cdot 40 \cdot 3 + 1 \cdot 35 \cdot 3 \pmod{280} \\ &= 513 \pmod{280} \\ &= 233. \end{aligned}$$

If the initial number of tokens in r is 233, all three competition paths with the same resource place r and different in-arcs and out-arcs weights that are represented by the weight matrix $[\mathbf{W}^{(r)}] = \begin{bmatrix} 5 & 7 & 8 \\ 3 & 2 & 1 \end{bmatrix}$ satisfy the above restrictive condition at the same time. Note that there is a limitation

on $w_i^{in(r)}$ in Theorem 4.1 to make Eq. (1) solvable by using Eq. (2). The limitation demands that $w_1^{in(r)}$, $w_2^{in(r)}$, \dots , and $w_n^{in(r)}$ should be pairwise relatively prime and greater than one. However, it does not always hold that $w_1^{in(r)}$, $w_2^{in(r)}$, \dots , and $w_n^{in(r)}$ are pairwise relatively prime in this context. For example, $w_1^{in(r)} = 2$ and $w_2^{in(r)} = 4$ are permitted and a solution also needs to be calculated for a weight matrix containing them. In addition, a solution obtained by using Eq. (2) makes every remainder to be exactly equal to $w_i^{out(r)}$. This is not necessary and always makes $M_0(r)$ relatively large. Therefore, the initial number of tokens in r , i.e., $M_0(r)$, is calculated by solving the following linear programming problem instead of using Eq. (2).

$$\left\{ \begin{array}{l} \min M_0(r) \\ s. t. \\ M_0(r) - k_1 \cdot w_1^{in(r)} \geq w_1^{out(r)} \\ M_0(r) - k_2 \cdot w_2^{in(r)} \geq w_2^{out(r)} \\ \dots \\ M_0(r) - k_n \cdot w_n^{in(r)} \geq w_n^{out(r)} \end{array} \right. \quad (4.3)$$

where $k_i = \lfloor \frac{M_0(r)}{w_i^{in(r)}} \rfloor$, $\frac{M_0(r)}{w_i^{in(r)}} - 1 < k_i \leq \frac{M_0(r)}{w_i^{in(r)}}$, $k_i \in \mathbb{N}$ and $i = (1, 2, \dots, n)$.

Note that the computational complexity is NP-hard when a mathematical programming is adopted to analyze the relationship between arc weights and initial markings. However, the programming problem with a small number of constraints and variables can be solved in very short time. For weight matrix $[\mathbf{W}^{(r)}] = \begin{bmatrix} 5 & 7 & 8 \\ 3 & 2 & 1 \end{bmatrix}$, by Eq. (3), we have $M_0(r) = 9$ such that $9 \bmod 5 \geq 3$, $9 \bmod 7 \geq 2$, and $9 \bmod 8 \geq 1$ hold at the same time. Note that the above analyses actually discuss a pairwise competitive relation of holders, i.e., the upstream and downstream activity places of every competition path, and the resulting imposed restriction. Therefore, the competition relationship among all upstream and downstream activity places of all competition paths with the same resource place cannot be ignored during the evolution analysis of a WS³PR. Similar to imposing a restrictive condition on a single competition path, the idea when considering all competition paths with the same resource place is that even if all upstream activity places are given a priority to maximally acquire the tokens in r , the remaining ones that cannot be obtained by any upstream activity place, are still adequate for at least one downstream activity place to complete an operation. On the whole, all restrictive conditions discussed above are embodied as Restriction 3.1 presented in

Chapter 3.

Note that in the following part of this chapter, it has the same meaning when we say that a resource place r in a competition path $l_{cp} = t_i r t_h$ satisfies Restriction 3.1 or a WSDC containing l_{cp} satisfies Restriction 3.1. Restriction 3.1 is an imposed limitation on the allocation of a resource place and its multiple competition paths. Since every WSDC contains at least one competition path, this restriction of a resource place is thus imposed on the WSDC containing this resource place.

According to Theorem 3.2, the numerical relationships among weights, and between weights and initial markings are investigated to ensure whether it is possible that every WSDC satisfies Restriction 3.1 at some particular initial markings. If so, an initial marking is then calculated through a set of developed algorithms in Chapter 3 and solving the corresponding linear programming problem. Therefore, the liveness-enforcement for a WS³PR is achieved through checking the structure of a net model, calculating and configuring a proper initial marking for resource places. Different from the siphon(monitor)-based liveness-enforcing methods, the approach further developed in this chapter unveils and takes a full advantage of the intrinsically live structure of generalized Petri nets, which is hidden behind the arc weights.

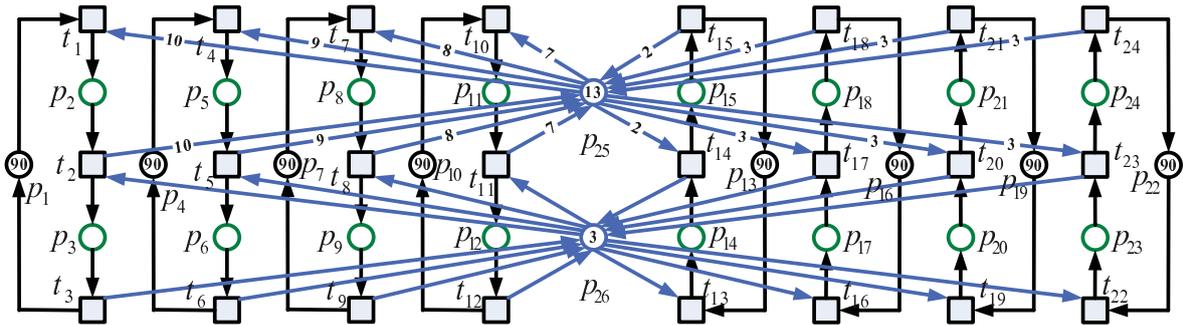


FIG. 4.3 – A more complex WS³PR net model with two highly shared resource places.

We also use the net model shown in Fig. 4.1(a) to illustrate the proposed method. According to the results in Chapter 3, the weighted digraph of resource places G_R is obtained as shown in Fig. 4.1(c). There are four WSDCs, $c_{w1} = t_2 p_{13} t_{11} p_{14} t_2$, $c_{w2} = t_3 p_{14} t_{10} p_{15} t_3$, $c_{w3} = t_4 p_{15} t_9 p_{16} t_4$, and $c_{w4} = t_5 p_{16} t_8 p_{17} t_5$, contained in G_R . A set of all resource paths is $\mathbf{L} = \{t_2 p_{13} t_{11}, t_{11} p_{14} t_2, t_3 p_{14} t_{10},$

$t_{10}p_{15}t_3, t_4p_{15}t_9, t_9p_{16}t_4, t_5p_{16}t_8, t_8p_{17}t_5\}$, and a set of all resource paths that can never satisfy Restriction 3.1 at any initial marking is $\mathbf{L}_F^* = \{t_{11}p_{14}t_2, t_{10}p_{15}t_3, t_9p_{16}t_4, t_8p_{17}t_5\}$. No WSDC can be formed by resource paths in \mathbf{L}_W^* , i.e., every WSDC contains at least one competition path satisfying Restriction 3.1. A minimal proper initial marking is calculated, and the net model is live at initial marking $M_0(P_R) = 3p_{13} + 5p_{14} + 3p_{15} + 7p_{16} + p_{17}$. It is worth noting that the reconfigured live net model with less resource capacity has totally 10147 states, much more than 4421 of the monitor-controlled system discussed in the previous subsection.

The WS³PR net model shown in Fig. 4.3 is a large example with two highly shared resource places. Since every WSDC contains resource place p_{25} , it is enough to consider only competition paths with p_{25} . Note that $p_2, p_5, p_8,$ and p_{11} are upstream activity places of p_{25} . $p_{15}, p_{18}, p_{21},$ and p_{24} are downstream activity places of p_{25} . However, every competition path with p_{25} represents a competition relationship between only an upstream activity place and a downstream one. The weight matrix with respect to p_{25} is $[\mathbf{W}^{(p_{25})}] = \begin{bmatrix} 10 & 10 & 9 & 9 & 8 & 8 & 7 & 7 \\ 3 & 2 & 3 & 2 & 3 & 2 & 3 & 2 \end{bmatrix}$. Every column of this matrix represents the in-arc and out-arc weights of a competition path with p_{25} . It is easy to find that all competition paths with p_{25} satisfy Restriction 3.1 if $M_0(p_{25}) = 13$, i.e., $(13 \bmod w_i^{in(p_{25})}) \geq w_i^{out(p_{25})}, i = 1, 2, 3, \dots, 8$, and there exists no such a vector \mathbf{A} such that $0 \leq (13 - \mathbf{A}[10, 9, 8, 7]^T) < 2$. This WS³PR is live at this initial marking.

As well known, deadlocks in a system are caused by the shortage of resources and their improper allocation. For the first reason, a straightforward method to avoid deadlocks is simply to increase the number of resources demanded by different competing processes. However, it is also unadvisable to blindly increase them without an insight into the numerical relationship between resource capacity and their multiple occupations by different processes. Take the aforementioned model in Fig. 4.1(a) as an example. It is not live at a new initial marking $M'_0(P_R) = 6p_{13} + 9p_{14} + 6p_{15} + 10p_{16} + p_{17}$ by simply adding more tokens into resource places. However, it is live at $M''_0(P_R) = 5p_{13} + 7p_{14} + 5p_{15} + 9p_{16} + p_{17}$ that also adds some tokens in resource places, and most importantly, M''_0 makes all WSDCs satisfy Restriction 3.1. On the contrary, it is possible to prevent deadlocks by decreasing the given resources to meet Restriction 3.1 according to this method. The net model is live at the minimal proper initial marking calculated above, i.e., $M_0(P_R) = 3p_{13} + 5p_{14} + 3p_{15} + 7p_{16} + p_{17}$. Of course, we should note that less resources mean less

system behavior. Also, distinguished from our work in previous chapters, the proposed method does not limit token counts in idle places, which model the numbers of different workpieces to be concurrently processed in the system. The largest number of all kinds of workpieces that can enter the processing system is equal to the sum of all activity places' bounds.

Clearly, the control method without adding any monitor has evident advantages over existing ones. First, it avoids the problem of structural complexity of a liveness-enforcing monitor-based supervisor that needs considerable control places and arcs. As a result, we significantly save the deadlock controller design efforts and implementation cost. Second, this method does not change the class of controlled models, in other words, the original plant model and controlled one both belong to the same subclass of Petri nets. Therefore, we can use the properties of the same subclass of Petri nets to analyze, verify, evaluate, and compare the before-and-after controlled systems. Third, the method is clearer for system designers and engineers to prevent system deadlocks by increasing or decreasing system resources according to some rules than adding monitors that may require additional hardware and/or software (costs). If the method is applied during the design stage, an inherently live model can be directly built.

However, we cannot shy away from the shortcoming of this deadlock prevention and liveness-enforcing method. It depends on a characteristic structure, i.e., all WSDCs have to satisfy some structural conditions, which greatly limits its application scope. Thus, we can innovatively combine this method with the existing ones based on siphons to advance the state of the art.

4.3 Hybrid liveness-enforcing policy

This section gives a hybrid liveness-enforcing method that combines the existing siphon control and characteristic structure further developed in the previous section. Note that a siphon-based liveness-enforcing method is not exclusively restricted to elementary siphons, and any other siphon (monitor)-based policy can be adopted in the proposed hybrid liveness-enforcing method. An example is used here to illustrate this method. A WS³PR net model is shown in Fig. 4.4. It totally has 8724 states among which 36 are deadlocks.

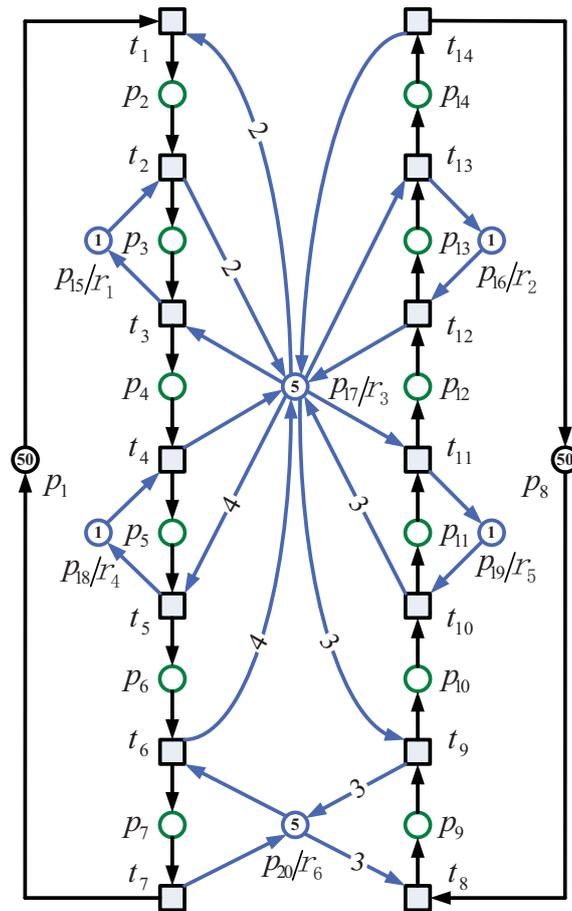


FIG. 4.4 – A WS³PR net model with 6 resource places.

4.3. HYBRID LIVENESS-ENFORCING POLICY

There are five WSDCs in its corresponding weighted digraph of resource places, $c_{W1} = t_2p_{17}t_3p_{15}t_2$, $c_{W2} = t_4p_{17}t_5p_{18}t_4$, $c_{W3} = t_{12}p_{17}t_{13}p_{16}t_{12}$, $c_{W4} = t_{10}p_{17}t_{11}p_{19}t_{10}$, and $c_{W5} = t_6p_{17}t_9p_{20}t_6$. In addition, there are 31 strict minimal siphons in this net model :

$$S_1^* = \{p_7, p_{14}, p_{15}, p_{16}, p_{17}, p_{18}, p_{19}, p_{20}\},$$

$$S_2^* = \{p_7, p_{12}, p_{14}, p_{15}, p_{17}, p_{18}, p_{19}, p_{20}\},$$

$$S_3^* = \{p_7, p_{10}, p_{14}, p_{15}, p_{16}, p_{17}, p_{18}, p_{20}\},$$

$$S_4^* = \{p_7, p_{10}, p_{12}, p_{14}, p_{15}, p_{17}, p_{18}, p_{20}\},$$

$$S_5^* = \{p_4, p_7, p_{14}, p_{15}, p_{16}, p_{17}, p_{19}, p_{20}\},$$

$$S_6^* = \{p_4, p_7, p_{12}, p_{14}, p_{15}, p_{17}, p_{19}, p_{20}\},$$

$$S_7^* = \{p_4, p_7, p_{10}, p_{14}, p_{15}, p_{16}, p_{17}, p_{20}\},$$

$$S_8^* = \{p_4, p_7, p_{10}, p_{12}, p_{14}, p_{15}, p_{17}, p_{20}\},$$

$$S_9^* = \{p_2, p_7, p_{14}, p_{16}, p_{17}, p_{18}, p_{19}, p_{20}\},$$

$$S_{10}^* = \{p_2, p_7, p_{12}, p_{14}, p_{17}, p_{18}, p_{19}, p_{20}\},$$

$$S_{11}^* = \{p_2, p_7, p_{10}, p_{14}, p_{16}, p_{17}, p_{18}, p_{20}\},$$

$$S_{12}^* = \{p_2, p_7, p_{10}, p_{12}, p_{14}, p_{17}, p_{18}, p_{20}\},$$

$$S_{13}^* = \{p_2, p_4, p_7, p_{14}, p_{16}, p_{17}, p_{19}, p_{20}\},$$

$$S_{14}^* = \{p_2, p_4, p_7, p_{12}, p_{14}, p_{17}, p_{19}, p_{20}\},$$

$$S_{15}^* = \{p_2, p_4, p_7, p_{10}, p_{14}, p_{16}, p_{17}, p_{20}\},$$

$$S_{16} = \{p_2, p_4, p_7, p_{10}, p_{12}, p_{14}, p_{17}, p_{20}\},$$

$$S_{17}^* = \{p_6, p_{14}, p_{15}, p_{16}, p_{17}, p_{18}, p_{19}\},$$

$$S_{18}^* = \{p_6, p_{12}, p_{14}, p_{15}, p_{17}, p_{18}, p_{19}\},$$

$$S_{19}^* = \{p_6, p_{10}, p_{14}, p_{15}, p_{16}, p_{17}, p_{18}\},$$

$$S_{20}^* = \{p_6, p_{10}, p_{12}, p_{14}, p_{15}, p_{17}, p_{18}\},$$

$$S_{21}^* = \{p_2, p_6, p_{14}, p_{16}, p_{17}, p_{18}, p_{19}\},$$

$$S_{22}^* = \{p_2, p_6, p_{12}, p_{14}, p_{17}, p_{18}, p_{19}\},$$

4.3. HYBRID LIVENESS-ENFORCING POLICY

$$S_{23}^* = \{p_2, p_6, p_{10}, p_{14}, p_{16}, p_{17}, p_{18}\},$$

$$S_{24} = \{p_2, p_6, p_{10}, p_{12}, p_{14}, p_{17}, p_{18}\},$$

$$S_{25}^* = \{p_4, p_6, p_{14}, p_{15}, p_{16}, p_{17}, p_{19}\},$$

$$S_{26}^* = \{p_4, p_6, p_{12}, p_{14}, p_{15}, p_{17}, p_{19}\},$$

$$S_{27}^* = \{p_4, p_6, p_{10}, p_{14}, p_{15}, p_{16}, p_{17}\},$$

$$S_{28} = \{p_4, p_6, p_{10}, p_{12}, p_{14}, p_{15}, p_{17}\},$$

$$S_{29} = \{p_2, p_4, p_6, p_{14}, p_{16}, p_{17}, p_{19}\},$$

$$S_{30} = \{p_2, p_4, p_6, p_{12}, p_{14}, p_{17}, p_{19}\}, \text{ and}$$

$$S_{31} = \{p_2, p_4, p_6, p_{10}, p_{14}, p_{16}, p_{17}\}.$$

Among the strict minimal siphons, $\{S_{16}, S_{24}, S_{28}, S_{29}, S_{30}, S_{31}\}$ can be chosen as a set of elementary siphons. Given a siphon S and a WSDC $c_W = (P_R^C, T_R^C, F_R^C, W_R^C)$, it is said that the siphon S contains c_W if $P_R^C \subseteq S$.

Theorem 4.2

A WS^3PR is live if all WSDCs contained in every siphon satisfy Restriction 3.1.

Proof : Since there exists no WSDC that is not contained in a siphon in WS^3PR , it means all WSDC in a WS^3PR satisfy Restriction 3.1. The rest proof is the same with that of Theorem 3.2.

Actually, Theorem 4.2 that links liveness, siphons, and WSDC together, is an alternative expression of Theorem 3.2 that enforces liveness purely from the viewpoint of WSDC instead of siphons. In Chapter 3, the relationship among liveness, structures, and initial markings is systematically studied by the concept of WSDC instead of siphons. However, siphons are a well-accepted structural object of Petri nets. This theorem attempts to build up links among siphons, a characteristic structure expressed by WSDC, and initial markings in a WS^3PR . In fact, the component formed by all WSDCs contained in a siphon is the structure of a CW of resource places [52] [64]. The next theorem binds WSDCs with siphons and further reveals their impact on the liveness of a WS^3PR . It makes the proposed hybrid liveness-enforcing method feasible by combining the traditional ones based on siphon control and the one based on a class of intrinsically live structures.

Theorem 4.3

A siphon S in a marked WS^3PR is never insufficiently marked if all WSDCs contained in it satisfy Restriction 3.1.

Proof : As aforementioned, there must be a downstream activity place p_{down} of a competition path $l_{cp} = t_r t_h$ belonging to $H_+([c_W])$ and thus belonging to a siphon S containing the CW $[c_W]$, i.e., $r, p_{down} \in S$. As for S , considering any marking $M \in R(N, M_0)$, there are two cases only.

Case (1) : If there are tokens remaining in the resource place r , $M(r) \geq \min_{r^*}$ always holds due to the second condition of Restriction 3.1.

Case (2) : If no token remains in the resource place r , there must be adequate tokens allocated in the downstream activity place p_{down} , and $\forall t \in p_{down}^{\bullet}, M(p_{down}) \geq W(p_{down}, t) = 1$ holds. If the downstream operation is completed, these tokens in p_{down} will return to r , and constitute an insufficient quantity to be acquired by any upstream activity place due to the imposed restrictions. In a word, S is never insufficiently marked at any reachable marking.

According to elementary siphons [58], the liveness-enforcement to the net shown in Fig. 4.4 is implemented by adding six monitors for six elementary siphons. However, the theory of elementary siphons does not consider the characteristic structure represented by WSDC and the relationship between arc-weights and initial markings. Consequently, Theorem 4.3 builds a bridge connecting siphons and WSDC, and improves elementary siphon-based control methods. Theorem 4.3 can be used to identify redundant monitors from those obtained by elementary siphons. In the example, six monitors are originally calculated according to elementary siphons as follows.

$$\bar{S}_{16} = 4p_6 + 3p_9, M_0(V_{S16}) = 10 - 6 = 4;$$

$$\bar{S}_{24} = p_4 + p_5, M_0(V_{S24}) = 6 - 4 = 2;$$

$$\bar{S}_{28} = 2p_2 + p_3, M_0(V_{S28}) = 6 - 4 = 2;$$

$$\bar{S}_{29} = 3p_{10} + p_{11} + p_{12} + p_{13}, M_0(V_{S29}) = 7 - 4 = 3;$$

$$\bar{S}_{30} = 3p_{10} + p_{11}, M_0(V_{S30}) = 6 - 3 = 3; \text{ and}$$

$$\bar{S}_{31} = p_{12} + p_{13}, M_0(V_{S31}) = 6 - 4 = 2.$$

The controlled net model with all six monitors is live but has 405 states only. Obviously,

compared with the number of all reachable states of the original net model without any control, the external monitors imposed to the net according to elementary siphons greatly restrict the system's behavior. Applying Theorem 4.3 to this example, we can find that three WSDCs, c_{W5} , c_{W1} , and c_{W4} contained by elementary siphons S_{16} , S_{28} , and S_{30} , respectively, all satisfy Restriction 3.1. Conversely, c_{W2} and c_{W3} contained by S_{24} and S_{31} respectively, do not satisfy it. Although c_{W4} contained by S_{29} satisfies it, another WSDC c_{W3} contained by S_{29} does not. Consequently, some monitors originally designed can be removed from the controlled net model, i.e., V_{S16} , V_{S28} , and V_{S30} are redundant. Their removal leads to a new controlled net model that is live and has 1194 states, which almost triples the state count in the prior controlled net based on all six elementary siphons. Its structural complexity is also reduced.

To sum up, the WSDC-based method is first used to check the whole structure of a WS^3PR and decide whether it can be controlled without any external monitors and then achieves the system liveness by configuring proper initial markings of resource places. If the WSDC-based method cannot be used to achieve the goal, an established siphon-based method can be adopted to design monitors. Before adding all obtained monitors to the plant model, Theorem 4.3 is applied to check every siphon for which a monitor has to be designed. The siphons that contain the characteristic structure are identified and classified into two groups. The first includes the siphons sufficiently marked at the current initial marking according to Theorem 4.3. The monitors for the siphons in this group are redundant and should be removed. The other group includes the siphons that are insufficiently marked at the current initial marking but will be sufficiently marked at some other proper initial markings. Their controllability can be achieved by adding monitors, or instead, re-configuring the initial numbers of tokens in the resource places contained by them. The final design of system supervisor will be a trade-off between the above two classes of methods according to the practical situation and control costs. The proposed hybrid policy aims to expand the application scope of a WSDC-based method, simplify the structure of a controlled system, and reduce corresponding control cost. For many practical cases, the WSDC-based method cannot replace siphon-based ones reported in [55], [60], and [62]. However, it can be mixed with them, thereby yielding an effective hybrid liveness-enforcing strategy.

Besides the sufficiently marked siphons discussed above, this chapter also gives results about

the minimal control of a siphon, which is not included in the existing studies. A majority of deadlock control and liveness enforcing methods for generalized Petri nets are designed on the ground of the maximal control of siphons. In contrast, the following two theorems present that a siphon or WS³PR net model is minimally controlled when all WSDCs contained by it satisfy Restriction 3.1.

Theorem 4.4

A siphon in a marked WS³PR is minimally controlled if all WSDCs contained in it satisfy Restriction 3.1.

Proof : Similar to Theorem 4.3.

A distinguished difference between ordinary and generalized Petri nets is that a siphon in a generalized net is not necessarily fully empty when a deadlock occurs. The numerical relationship between arc multiplicity and different initial markings must be thoroughly considered in the design and application of any deadlock control and liveness-enforcing policy for systems modeled by generalized Petri nets. Maximal and minimal controls are two important thoughts on siphon control. Compared with each other, the former is more conservative than the latter. Theorem 4.4 guarantees that a siphon whose all WSDCs satisfy Restriction 3.1 is intrinsically minimally controlled, i.e., $\exists p \in S, M(p) \geq \min_{p^*}, \forall M \in R(N, M_0)$.

Theorem 4.5

A marked WS³PR is minimally controlled and live if all WSDCs contained in every siphon satisfy Restriction 3.1.

Proof : It follows from Theorems 4.2, 4.3, and 4.4.

Theorems 4.4 and 4.5 bridge the liveness, minimal control, WSDC structures, and proper initial markings together. For example, considering the model shown in Fig. 4.1(a), every strict minimal siphon is minimally controlled since all WSDCs satisfy Restriction 3.1, and the net model is thus intrinsically minimally controlled and live at the initial marking $M_0(P_R) = 3p_{13} + 5p_{14} + 3p_{15} + 7p_{16} + p_{17}$. Note that $M_0(p_1)$ and $M_0(p_7)$ can be any positive numbers.

The results presented in this section build a bridge to make siphon(monitor)-based liveness-enforcing methods meet a WSDC-based one. They can be combined together to generate hybrid ones that can reduce the structural complexity and allow more system states.

4.4 Summary

This chapter presents a hybrid liveness-enforcing policy that combines the well established siphon(monitor)-based methods together with the recently proposed and enhanced WSDC-based one. The latter fully facilitates a characteristic structure of WS^3PR to lay a foundation of a system's self-controllability at some particular initial markings. To this end, the numerical relationships among arc-weights and between weights and initial markings are thoroughly investigated and utilized to enforce liveness. The hybrid policy enlarges the application scope of the WSDC-based method, and further improves the siphon(monitor)-based ones by identifying and removing the redundant monitors, which jumble in all control places obtained by the current methods. This reduces the structural complexity of the controlled system and allows more system states, and in turn saves the system control cost. However, there are still some open issues for future research. First, the internal mechanism of the WSDC-based method should be studied further to conceive a novel methodology for designing external monitors. Second, the application of the hybrid method should be extended to other generalized Petri nets. This requires to extend the concept of WSDC. Third, since a set of elementary siphons is not unique, determining how to choose an optimal set of elementary siphons while considering their structures and initial numbers of tokens is an important direction for the future work. The formalism is also expected to be checked for hybrid dynamical systems.

4.4. SUMMARY

Chapitre 5

Parameterized Liveness and Ratio-enforcing Supervisor for WS^3PR

This chapter proposes a synthesis method of supervisors for flexible manufacturing systems modeled by WS^3PR . A concept of resource usage ratios (RU-ratios) is first presented to describe the occupation degree of a resource by an operation. Next, an intrinsically live structure (ILS) characterized by a special numerical relationship between arc-weights and initial markings is investigated from a perspective of RU-ratios. Then, a new kind of supervisors is synthesized on the ground of the generic nature of ILSs. Such a supervisor can achieve the purposes of both liveness-enforcement and resource usage ratio-enforcement of the system under consideration. Given a plant, it is easy to determine the topological structure of such a supervisor and the number of monitors is bounded by that of resources used in the plant. In addition, when the configuration of the plant model changes, the supervisor can be reusable by adjusting control parameters only without the rearrangement of connections. This makes it easy enough and intuitive to be used by industrial practitioners. Instead of maximal behavioral permissiveness, it pursues a precise usage of shared resources that are limited and valuable.

5.1 Introduction

In Chapters 2 and 3, the structure of WS^3PR is explored. A new structural object, called weighted simple directed circuits (WSDCs), is defined to describe the structure of augmented circular

waits (CWs) among resources, instead of the traditionally used siphons. A siphon as a subset of places does not carry the weight information of arcs. The work in the previous chapters focuses on the competition relationship between an upstream activity place and a downstream one of a competition path contained by a WSDC. A restriction is proposed to identify this kind of ILSs and develop a WSDC-based method that enforces liveness by re-configuring the initial marking of a plant model. Without external monitors, the liveness-enforcement of a WS³PR is achieved by taking the full advantage of ILSs in Chapter 3. However, the requirement for this kind of global special structures limits the application scope of the WSDC-based method. Therefore, in Chapter 4 we combine it with elementary siphons [61] [58] to generate a hybrid liveness-enforcing policy that can simplify the supervisor for a system, improve the permissiveness, and finally, reduce the control implementation cost. In addition, the min-marked and minimally controlled siphons are also analyzed with the assistance of WSDC.

Since the seminal work by Ezpeleta *et al.* [16] is published, a variety of deadlock prevention and liveness-enforcing methods are implemented by adding monitors to an original plant model, most of which are designed to achieve siphon control. In this chapter, a new kind of monitors is proposed to control resource places directly. A so-called control path containing such a monitor is imposed on a plant model and actually replaces a competition path containing a resource place to be controlled. This results in a new WSDC containing the control path that is designed to satisfy a proposed restriction. The behavior of the original WSDC that contains the competition path is manipulated by the new one. In order to design such a monitor, a concept of RU-ratios is first proposed to describe the occupation degree of a resource by an operation. A competition for a resource between two operations is expressed by a relationship between their RU-ratios. Next, the generic nature of ILSs is analyzed from a viewpoint of the RU-ratio. Then, the ratio information and a numerical restriction are used to synthesize a liveness and ratio-enforcing supervisor.

The new supervisor proposed in the chapter achieves liveness and ratio-enforcement at the same time. In particular, the liveness-enforcement is implemented by imposing a proper resource usage ratio-enforcement on a plant model. It is an attempt to pursue a precise usage of resources instead of maximal behavioral permissiveness. The number of monitors in such a supervisor is bounded by that of resources used in the plant model. Hence, the control cost can be estimated wi-

thout difficulty. In addition, the topology of the supervisor can be easily determined when a plant model is given, and is simple enough to be understood and adopted by industrial practitioners. When the initial configuration and weights of the plant model are changed, the proposed supervisor can adapt quickly by adjusting control parameters only. The connection of monitors remains unchanged. It is just the work style of today’s programable logic controllers, i.e., the re-hard-wired job can be avoided. This greatly saves control cost.

5.2 Liveness and ratio-enforcement in WS³PR

5.2.1 Resource usage ratio

From the perspective of resource allocation, FMS modeled with Petri nets can be classified into three categories according to the number and type of resource units required when a workpiece is processed at an operation step : (1) only one unit of a single type of resources, (2) one or multiple units of a single type of resources, and (3) one or multiple units of one or multiple types of resources. Since the weights of the arcs connecting activity and resource places can be greater than one, an FMS modeled by WS³PR falls into the second category. That is to say, in a WS³PR, one or more tokens of only one resource place are used to complete a process of a workpiece represented by a token in an activity place.

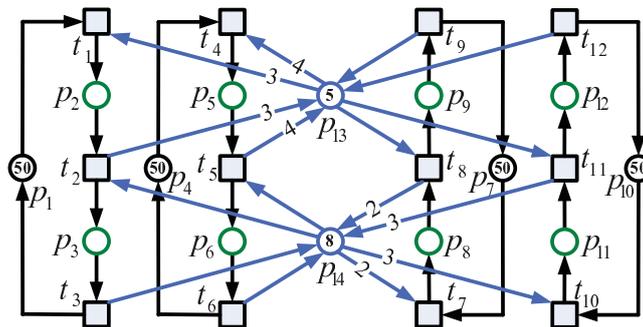


FIG. 5.1 – A WS³PR net with two shared resource places.

For example, in the WS³PR net shown in Fig. 5.1, when a token is in activity place p_2 , three tokens from resource place p_{13} will be used to complete the operation. Because of the characteristic

resource requirement of WS³PR, it is necessary to investigate a ratio of the number of tokens in an activity place to the initial number of tokens in a resource place that is used by the activity place. The number of tokens in a holder represents the number of workpieces processed in this operation and implies the number of tokens acquired from the resource place. The initial number of tokens in the resource place means the capability of this type of resources to be used by their holders. Thus, as a behavioral indicator, the ratio reflects an occupation degree of the resource place by one of its holders at a reachable state. Its formal definition is given as follows.

Definition 5.1 In a marked WS³PR (N, M_0) with $N = (P_0 \cup P_A \cup P_R, T, F, W)$, a ratio $\lambda^{r \rightarrow p} = \frac{M(p) \cdot W(r, t)}{M_0(r)}$, is called a resource usage ratio (RU-ratio) of activity place p with respect to resource place r at M if $p \in H(r)$, where $r \in P_R, t \in \bullet p \cap r^\bullet$ and $M \in R(N, M_0)$.

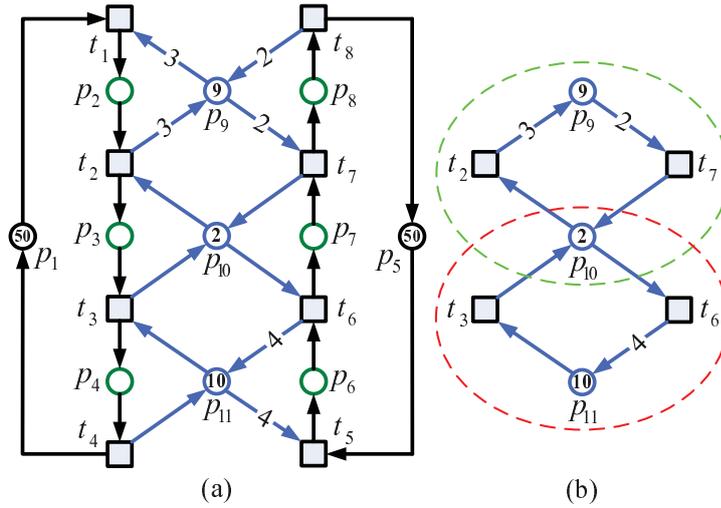


FIG. 5.2 – (a) A generalized net and (b) one of its subnets.

For example, in the WS³PR net shown in Fig. 5.2(a), $M = 39p_1 + 2p_2 + p_3 + 8p_4 + 49p_5 + p_7 + 3p_9 + 2p_{11}$ is a reachable state. The RU-ratios of activity places p_2 and p_4 with respect to resource places p_9 and p_{11} at M are $\lambda^{p_9 \rightarrow p_2} = (2 \cdot 3)/9 = 2/3$ and $\lambda^{p_{11} \rightarrow p_4} = (8 \cdot 1)/10 = 4/5$, respectively. They reflect resource places p_9 and p_{11} 's current occupation rates of activity places p_2 and p_4 at M . $\lambda^{p_9 \rightarrow p_8} = \lambda^{p_{11} \rightarrow p_6} = 0$ at M means that p_6 and p_8 do not use any token of p_{11} and p_9 , respectively. $\lambda^{p_{10} \rightarrow p_3} = \lambda^{p_{10} \rightarrow p_7} = 1/2$ implies that both p_3 and p_7 use half capacity of p_{10} at M .

An RU-ratio of activity place p with respect to resource place r depends on markings. The maximal RU-ratio of p with respect to r , denoted by $\max\lambda^{r \rightarrow p}$, is a ratio of $\max M(p) \cdot W(r, t)$ to $M_0(r)$, where $\max M(p)$ is the maximal number of tokens in p , $\max M(p) \cdot W(r, t)$ is the maximal number of tokens that p can exclusively acquire from r , $p \in H(r)$, and $t \in \bullet p \cap r^\bullet$. Note that $\max M(p)$ is different from the bound of the activity place p , which is the maximal number of tokens in p among all reachable states of $R(N, M_0)$. It is decided by the initial number of tokens in resource place r and the weight of the arcs connecting its holder p and r , i.e., $\max M(p) = \lfloor \frac{M_0(r)}{W(r, t)} \rfloor$, the largest integer less than or equal to $\frac{M_0(r)}{W(r, t)}$. It is also the maximal number of continuous firings of transition t . It is not related to the initial markings in process idle places.

All possible numbers of tokens in p are sorted in a descending order, $\lfloor \frac{M_0(r)}{W(r, t)} \rfloor, \lfloor \frac{M_0(r)}{W(r, t)} \rfloor - 1, \lfloor \frac{M_0(r)}{W(r, t)} \rfloor - 2, \dots, 2, 1$, and 0. Consequently, all possible values of $\lambda^{r \rightarrow p}$ are a series of discrete numbers with a general expression $\lambda_i^{r \rightarrow p} = \frac{(\lfloor \frac{M_0(r)}{W(r, t)} \rfloor - i) \cdot W(r, t)}{M_0(r)}$, $i \in \{0, 1, 2, \dots, \lfloor \frac{M_0(r)}{W(r, t)} \rfloor\}$. In the rest of the chapter, for economy of space, we use $[\lambda_i^{r \rightarrow p}, 0]^{r \rightarrow p}$ to express an admissible range of RU-ratios that includes any possible value between $\lambda_i^{r \rightarrow p}$ and 0 (including $\lambda_i^{r \rightarrow p}$ and 0)¹. Since $\lambda_0^{r \rightarrow p} = \max\lambda^{r \rightarrow p}$, $[\lambda_0^{r \rightarrow p}, 0]^{r \rightarrow p} = [\max\lambda^{r \rightarrow p}, 0]^{r \rightarrow p}$ is a set of all possible values of an RU-ratio of p with respect to r .

For example, in the WS³PR net shown in Fig. 5.2(a), $\max M(p_4) = \lfloor \frac{10}{1} \rfloor = 10$ at any initial marking of process idle places. However, the bound of p_4 is 1 if $M_0(p_1) = 1$. The maximal number of tokens in an activity place is decided only by the structure and the initial number of tokens in a resource place that is used by the activity place. Note that it cannot be reached if the process idle place associated with the activity place is not sufficiently marked. In Fig. 2.2(a), $\max\lambda^{p_{11} \rightarrow p_4} = \frac{\max M(p_4) \cdot W(p_{11}, t_3)}{M_0(p_{11})} = \frac{10 \cdot 1}{10} = 1$ means that the operation represented by activity place p_4 can completely occupy all capabilities of this resource represented by resource place p_{11} . All possible token counts in p_4 are 10, 9, 8, ..., 2, 1, and 0. They are consecutive integers. Thus, all possible values of $\lambda^{p_{11} \rightarrow p_4}$ sorted in a descending order are 1, 9/10, 8/10, ..., 2/10, 1/10, and 0. Note that it is the same situation as in an S³PR since all arc weights are equal to one. It implies that the units of a resource can be used by an operation one by one increasingly until all units are occupied.

¹In order to be clear, all values in the set are sorted in a descending order.

$\max\lambda^{p_9 \rightarrow p_2}$ is also equal to one. The fact that $M_0(p_9) = 9$ is divisible by $W(p_9, t_1) = 3$ means that all tokens in p_9 can be used by p_2 three by three increasingly until p_9 is empty. Both operations modeled by p_2 and p_4 may occupy all capacities of resources modeled by p_9 and p_{11} , respectively. Different from them, $\max\lambda^{p_{11} \rightarrow p_6} = 8/10$ means that even allowing p_6 to maximally require tokens from p_{11} , two tokens, i.e., $2/10$ capacity of this resource, will remain in p_{11} . It is determined by the numerical relationship between $M_0(p_{11})$ and $W(p_{11}, t_5)$, i.e., $M_0(p_{11})$ cannot be divisible by $W(p_{11}, t_5)$, and the remainder is 2. Full details about RU-ratios of every activity place with respect to its resource place are given in Table 7.1.

In summary, if all tokens in a resource place in S³PR are allowed to be exclusively used by one of its holders, they can be completely acquired by this activity place. However, since the arc weights in WS³PR may be greater than one, some activity places cannot occupy all tokens from their respective resource places. Therefore, the concept of resource usage ratios is proposed to investigate the use of tokens in a resource place by an activity place. Next, the aforementioned intrinsically live structure will be demonstrated from its viewpoint. The relationship between RU-ratios of upstream and downstream activity places is discussed with the concept of WSDC.

TABLE 5.1 – Full details about RU-ratios of WS³PR net in Fig. 5.2(a)

No.	r	p	$\max\lambda^{r \rightarrow p}$	all possible values	admissible range
1	p_9	p_2	1	1, 2/3, 1/3, 0	$[1, 0]^{p_9 \rightarrow p_2}$
2	p_9	p_8	8/9	8/9, 6/9, 4/9, 2/9, 0	$[8/9, 0]^{p_9 \rightarrow p_8}$
3	p_{10}	p_3	1	1, 1/2, 0	$[1, 0]^{p_{10} \rightarrow p_3}$
4	p_{10}	p_7	1	1, 1/2, 0	$[1, 0]^{p_{10} \rightarrow p_7}$
5	p_{11}	p_4	1	1, 9/10, 8/10, ..., 1/10, 0	$[1, 0]^{p_{11} \rightarrow p_4}$
6	p_{11}	p_6	8/10	8/10, 6/10, 4/10, 2/10, 0	$[8/10, 0]^{p_{11} \rightarrow p_6}$

5.2.2 Ratio analysis of an intrinsically live structure

As mentioned in the previous section, an ILS is a WSDC containing at least one competition path satisfying Restriction 3.1. In this subsection, it is analyzed from the viewpoint of the RU-ratio of an activity place with respect to its resource place involved in WSDC. In particular, the relationship between RU-ratios of upstream and downstream activity places of a competition path

is deliberated to develop a new kind of supervisors, which is presented later.

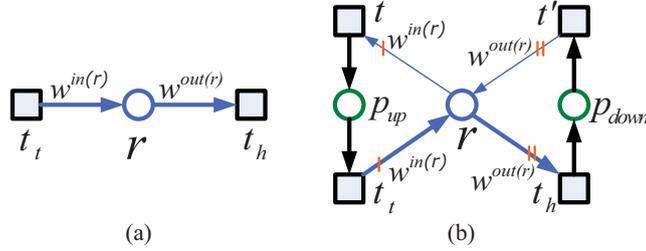


FIG. 5.3 – (a) A competition path and (b) a subnet containing the competition path and its upstream and downstream activity places.

A competition path $t_r t_h$ as shown in Fig. 5.3(a) reflects a pair-wised competition relationship between upstream activity place p_{up} and downstream one p_{down} shown in Fig. 5.3(b). The RU-ratios of p_{up} and p_{down} with respect to resource place r are denoted by $\lambda^{r \rightarrow p_{up}}$ and $\lambda^{r \rightarrow p_{down}}$, respectively. According to Definition 5.1, $\lambda^{r \rightarrow p_{up}} = \frac{M(p_{up}) \cdot W(r, t)}{M_0(r)} = \frac{M(p_{up}) \cdot w^{in(r)}}{M_0(r)}$ and $\lambda^{r \rightarrow p_{down}} = \frac{M(p_{down}) \cdot W(r, t_h)}{M_0(r)} = \frac{M(p_{down}) \cdot w^{out(r)}}{M_0(r)}$.

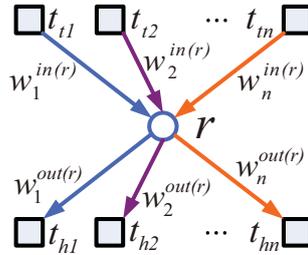


FIG. 5.4 – Multiple competition paths with a same resource place r .

If there are totally n competition paths with the same resource place r as shown in Fig. 5.4, which are expressed by a weight matrix $\mathbf{W}^{(r)}$, the corresponding RU-ratios of upstream and downstream activity places of $t_{ti} r t_{hi}$ with respect to r are $\lambda^{r \rightarrow p_{upi}}$ and $\lambda^{r \rightarrow p_{downi}}$, $i \in \mathbb{N}_n$.

For example, in the WS³PR shown in Fig. 5.5(a), $t_5 p_{10} t_8$, $t_8 p_{11} t_2$, $t_2 p_{10} t_8$, and $t_8 p_{11} t_5$ are all competition paths of the net model. p_5 and p_9 are upstream and downstream activity places of $t_5 p_{10} t_8$, respectively. The admissible ranges of $\lambda^{p_{10} \rightarrow p_5}$ and $\lambda^{p_{10} \rightarrow p_9}$ are $[1, 0]^{p_{10} \rightarrow p_5} = [1, 0]^{p_{10} \rightarrow p_9} =$

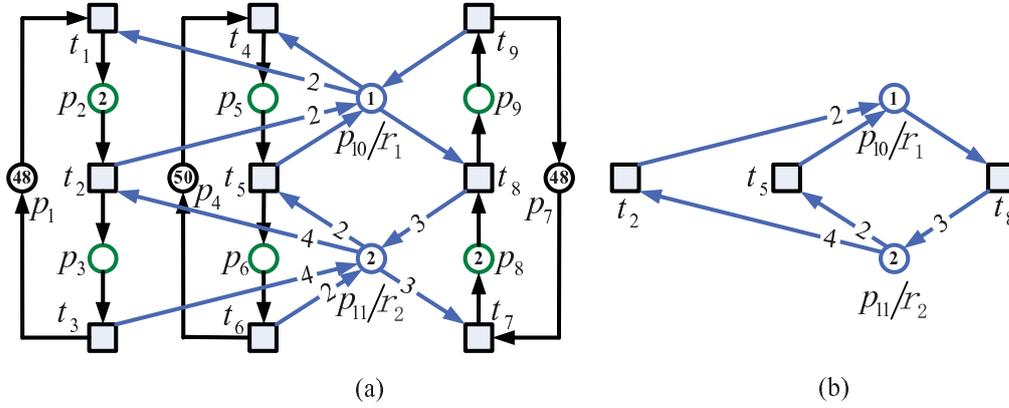


FIG. 5.5 – (a) A WS³PR net with initial marking $M_0 = 50p_1 + 50p_4 + 50p_7 + 5p_{10} + 8p_{11}$ and (b) two WSDCs, each of which contains a competition path satisfying Restriction 3.1.

$\{1, 4/5, 3/5, 2/5, 1/5, 0\}$. It means that p_5 and p_9 have exactly the same way to use p_{10} . All tokens in p_{10} may be monopolized by p_5 or p_9 . In contrast, p_8 and p_3 , the upstream and downstream activity places of $t_8p_{11}t_2$, respectively, have a different way to use p_{11} with $[1, 0]^{p_{11} \rightarrow p_3} = \{1, 1/2, 0\}$ and $[3/4, 0]^{p_{11} \rightarrow p_8} = \{6/8, 3/8, 0\}$. After a prioritized and maximal acquirement of tokens in p_{11} , the operation modeled by p_8 can maximally occupy $3/4$ capability of this resource. It is clear that p_8 can never acquire the remaining $1/4$ capability of p_{11} by the structure of the plant model and the initial marking. Moreover, $1/4$ is less than the smallest non-zero value in $[1, 0]^{p_{11} \rightarrow p_3}$, i.e., the remaining capability is not enough to be used by p_3 at the minimum level. This is exactly a situation from which condition (1) of Restriction 3.1 is trying to prevent.

This condition is a limitation on the numerical relationship between initial markings of the resource place and weights of the in-arc and out-arc of a single competition path. It ensures that all tokens in the resource place cannot be acquired only by the upstream activity place of the competition path. In terms of RU-ratios, $t_2p_{10}t_8$ and $t_8t_{11}p_5$ in Fig. 5.5(b) can be used to illustrate this condition positively. p_2 and p_9 (resp. p_8 and p_6) are the upstream and downstream activity places of competition path $t_2p_{10}t_8$ (resp. $t_8p_{11}t_6$), respectively. $[4/5, 0]^{p_{10} \rightarrow p_2} = \{4/5, 2/5, 0\}$, $[1, 0]^{p_{10} \rightarrow p_9} = \{1, 4/5, 3/5, 2/5, 1/5, 0\}$, $[9/11, 0]^{p_{11} \rightarrow p_8} = \{9/11, 6/11, 3/11, 0\}$ and $[1, 0]^{p_{11} \rightarrow p_6} = \{1, 6/8, 4/8, 2/8, 0\}$. When p_2 reaches its maximal RU-ratio, the remaining capability of resource place p_{10} , i.e., $1/5$, is in the admissible range of $\lambda^{p_{10} \rightarrow p_9}$, i.e., $[1, 0]^{p_{10} \rightarrow p_9}$. In other words, the RU-ratio of p_9 can maintain a non-zero value in the admissible range in case that

its competitor p_2 obtains the most tokens that are allowed by the structure and the initial marking. The same situation can be found in $t_8p_{11}t_5$. When the RU-ratio of p_8 reaches $\max\lambda^{p_{11} \rightarrow p_8} = 6/8$, the minimal non-zero value in the admissible range of $\lambda^{p_{11} \rightarrow p_6}$, i.e., $2/8$, can also be guaranteed. As shown in Fig. 5.5(b), there are two WSDCs contained in this weighted digraph of resource places, i.e., $t_2p_{10}t_8p_{11}t_2$ and $t_5p_{10}t_8p_{11}t_5$. Each contains a competition path satisfying Restriction 3.1.

The inequality in condition (1) of Restriction 3.1 can be explained in the perspective of RU-ratios of the upstream and downstream activity places with respect to the resource place of a competition path. Actually,

$$M_0(r) \bmod w^{in(r)} \geq w^{out(r)}$$

is equivalent to

$$M_0(r) - \lfloor \frac{M_0(r)}{w^{in(r)}} \rfloor \cdot w^{in(r)} \geq w^{out(r)}.$$

Every item is divided by $M_0(r)$, leading to

$$\frac{M_0(r)}{M_0(r)} - \frac{\lfloor \frac{M_0(r)}{w^{in(r)}} \rfloor \cdot w^{in(r)}}{M_0(r)} \geq \frac{w^{out(r)}}{M_0(r)}.$$

By conversions and substitutions, we have

$$1 - \frac{\max M(p_{up}) \cdot w^{in(r)}}{M_0(r)} \geq \frac{w^{out(r)}}{M_0(r)} = \frac{1 \cdot w^{out(r)}}{M_0(r)} = \frac{\min M(p_{down}) \cdot w^{out(r)}}{M_0(r)}.$$

This inequality can be simplified as follows.

$$1 - \max\lambda^{r \rightarrow p_{up}} \geq \min\lambda^{r \rightarrow p_{down}}$$

where $\min\lambda^{r \rightarrow p_{down}}$ is the minimal non-zero value in the admissible range of $\lambda^{r \rightarrow p_{down}}$.

The second condition of Restriction 3.1 can also be explained from the viewpoint of the RU-ratio. For example, in the WS³PR shown in Fig. 5.1, competition paths $t_2p_{13}t_8$, $t_2p_{13}t_{11}$, $t_5p_{13}t_8$ and $t_5p_{13}t_{11}$ are associated with the same resource place p_{13} . p_2 and p_5 (resp. p_9 and p_{12}) are

upstream (resp. downstream) activity places of resource one p_{13} in terms of the four competition paths. $[3/5, 0]^{p_{13} \rightarrow p_2} = \{3/5, 0\}$, $[4/5, 0]^{p_{13} \rightarrow p_5} = \{4/5, 0\}$, and $[1, 0]^{p_{13} \rightarrow p_9} = [1, 0]^{p_{13} \rightarrow p_{12}} = \{1, 4/5, 3/5, 2/5, 1/5, 0\}$. It can be found that if the tokens are first required and held by one of its upstream activity place, p_2 or p_5 , for completing one operation, the remaining capability of resource place p_{13} is no longer enough to be requested by p_2 or p_5 , but greater or equal to the minimal non-zero value of the admissible range of $\lambda^{p_{13} \rightarrow p_9}$ and $\lambda^{p_{13} \rightarrow p_{12}}$. In contrast with p_{13} , all the tokens in p_{14} can be taken by p_8 and/or p_{11} , and there is nothing left for the execution of each downstream activity operations, i.e., p_3 or p_6 . Therefore, the second condition of Restriction 3.1 can be expressed with the RU-ratio as follows :

$$1 - \sum_{i=1}^n \lambda^{r \rightarrow p_{upi}} \geq \min\{\min \lambda^{r \rightarrow p_{downi}}\}, \text{ if } 1 - \sum_{i=1}^n \lambda^{r \rightarrow p_{upi}} > 0,$$

where $\lambda^{r \rightarrow p_{upi}} \in [\max \lambda^{r \rightarrow p_{upi}}, 0]^{p_{upi}}$, $\lambda^{r \rightarrow p_{downi}} \in [\max \lambda^{r \rightarrow p_{downi}}, 0]^{p_{downi}}$, and n is the number of columns in the weight matrix representing n competition paths with the same resource place r .

Restriction 5.1

Given WS³PR (N, M_0) , a resource place r satisfies the following two conditions :

- (1) $1 - \max \lambda^{r \rightarrow p_{upi}} \geq \min \lambda^{r \rightarrow p_{downi}}$; and
- (2) $1 - \sum_{i=1}^n \lambda^{r \rightarrow p_{upi}} \geq \min\{\min \lambda^{r \rightarrow p_{downi}}\}$, if $1 - \sum_{i=1}^n \lambda^{r \rightarrow p_{upi}} > 0$;

where p_{upi} and p_{downi} are upstream and downstream activity places with respect to every competition path containing r , $\lambda^{r \rightarrow p_{upi}} \in [\max \lambda^{r \rightarrow p_{upi}}, 0]^{p_{upi}}$, $\lambda^{r \rightarrow p_{downi}} \in [\max \lambda^{r \rightarrow p_{downi}}, 0]^{p_{downi}}$, and n is the number of columns of weight matrix representing n competition paths with the same resource place r .

In brief, the intrinsically live structure represented by Restriction 3.1 comes down to a limitation on RU-ratios as expressed in Restriction 5.1. After a prioritized and maximal acquirement of tokens in a resource place by its one or multiple upstream activity places, the remaining capability of the resource place should be greater or equal to the minimal non-zero RU-ratio of at least one of its downstream places. It also helps ones design a new control method that aims at achieving the purposes of liveness and ratio-enforcement together.

5.2.3 Synthesis of liveness and ratio-enforcing supervisors

Inspired by ILSs and Restriction 3.1, a synthesis method of a new supervisor that blends liveness-enforcement with ratio-enforcement is proposed in this subsection. First, the topology of this kind of liveness and ratio-enforcing supervisors (LRS, for short) is given based on the plant model. Next, the parameter design of an LRS is solved with the tool of mathematical programming. Finally, the parameter adjustment of an LRS is investigated in order to implement different resource utilization ratios.

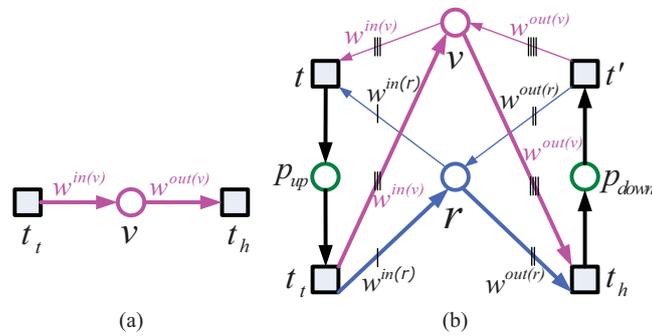


FIG. 5.6 – (a) A control path and (b) the control path and corresponding paired arcs imposed on a subnet containing a competition path.

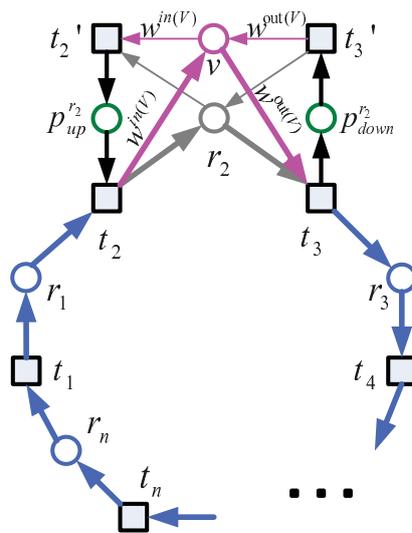


FIG. 5.7 – A general WSDC with an LRS monitor v for resource place r_2 .

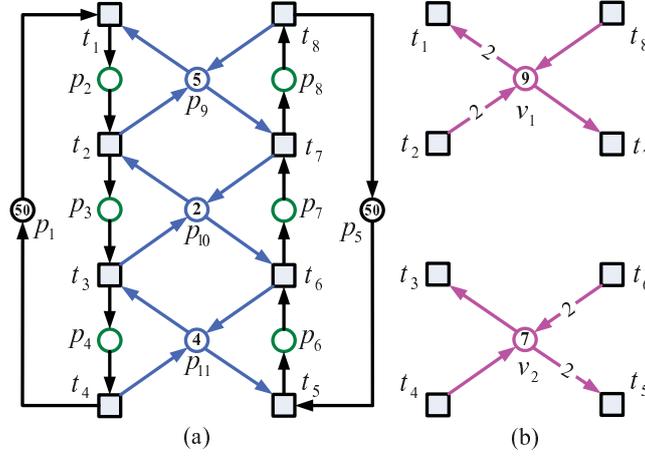
Like the supervisors proposed in literature, an LRS also consists of monitors (control places) and transitions. Similar to the role that a competition path satisfying Restriction 3.1 plays in a WSDC, an external control path shown in Fig. 5.6(a) is introduced to virtually replace a competition path of a plant model, which does not satisfy Restriction 3.1.

As shown in Fig. 5.7, an LRS monitor v is imposed on a general WSDC to control the competition path $t_2r_2t_3$. Comparing with the analysis sequence of ILS in Chapter 3, the construction sequence is reversed. It is from a resource place to a competition path to a control path and further to a WSDC. The essence of the content in Chapter 3 and this chapter is exactly the same. The only difference is that the former is a deconstruction and the latter is a construction.

Definition 5.2 Given a competition path $t_r t_h$ in a WS³PR, a simple path $\mathbb{I}^{\text{CD}} = t_v t_h$ is said to be a control path if v is an external control place. (t_r, v) and (v, t_h) are called the in-arc and out-arc of the control place v , and their weights are denoted by $w^{\text{in}(v)}$ and $w^{\text{out}(v)}$, respectively.

A well-designed control path satisfies Restriction 3.1 or Restriction 5.1 and is used to coordinate a competition relation represented by the competition path to be controlled. As shown in Fig. 5.6(b), when the paired arcs of the in-arc and the out-arc of $t_v t_h$ are supplemented into the plant model, we can find that the monitor v is connected with transitions of the plant model in the exactly same way with resource place r . Hence, being different from supervisors in literature, the topology of an LRS is determined when a plant model and competition paths to be controlled are given. For example, for the S³PR net model shown in Fig. 5.8(a), if the competition paths $t_2 p_9 t_7$ and $t_6 p_{11} t_3$ are chosen to be controlled, the topology of the corresponding LRS is determined, as shown in Fig.5.8(b). This alleviates the computation burden when designing an LRS.

In the S³PR shown in Fig. 5.8(a), the competition path $t_2 p_9 t_7$ represents a pair-wise competition relationship between an upstream activity place p_2 and a downstream one p_8 . $[1, 0]^{p_9 \rightarrow p_2} = [1, 0]^{p_9 \rightarrow p_8} = \{1, 4/5, 3/5, 2/5, 1/5, 0\}$ means that p_2 can fully possess all tokens in p_9 and none is left to make p_8 , and vice versa. Since the weights of the in-arc and out-arc of $t_2 p_9 t_7$ are equal to one, the current structure defies Restriction 3.1. In order to control this structure, the induced monitor v_1 corresponds to resource place p_9 and acts as a virtual resource place to limit the tokens allocation of p_9 through setting the numerical relationship between the initial number of tokens configured in v_1 and the weights of the in-arc (t_2, v_1) and out-arc (v_1, t_7) . The


 FIG. 5.8 – (a) An S³PR net model to be controlled and (b) an LRS synthesized for the net.

other competition path to be controlled is $t_6p_{11}t_3$. It neither satisfies Restriction 3.1. Note that $[4/5, 0]^{p_{11} \rightarrow p_4} = \{4/5, 2/5, 0\}$ and $[1, 0]^{p_{11} \rightarrow p_6} = \{1, 4/5, 3/5, 2/5, 1/5, 0\}$. Therefore, monitor v_2 and the corresponding arcs are used to control this competition path.

The initial marking of monitor v (resp. resource place r) and weights $w^{in(v)}$ and $w^{out(v)}$ (resp. $w^{in(r)}$ and $w^{out(r)}$) are called the parameters of control path t_tvt_h (resp. competition path t_trt_h). Formally, we use $\left\{ \begin{bmatrix} w^{in(v)} \\ w^{out(v)} \end{bmatrix}, M_0(v) \right\}$ and $\left\{ \begin{bmatrix} w^{in(r)} \\ w^{out(r)} \end{bmatrix}, M_0(r) \right\}$ to express a set of parameters of monitor v and resource place r with respect to a control path t_tvt_h and a competition path t_trt_h , respectively. For example, in Fig. 5.8(b), the current set of parameters of monitor v_1 with respect to the control path $t_2v_1t_7$ is $\left\{ \begin{bmatrix} w^{in(v_1)} \\ w^{out(v_1)} \end{bmatrix}, M_0(v_1) \right\} = \left\{ \begin{bmatrix} 2 \\ 1 \end{bmatrix}, 9 \right\}$, and that of monitor v_2 with respect to $t_6v_2t_3$ is $\left\{ \begin{bmatrix} w^{in(v_2)} \\ w^{out(v_2)} \end{bmatrix}, M_0(v_2) \right\} = \left\{ \begin{bmatrix} 2 \\ 1 \end{bmatrix}, 7 \right\}$.

In general, $\{[\mathbf{W}^{(v)}], M_0(v)\}$ and $\{[\mathbf{W}^{(r)}], M_0(r)\}$ are used to express a set of parameters of monitor v and resource place r with respect to n control paths and n competition paths, respectively, where $[\mathbf{W}^{(v)}] = \begin{bmatrix} \mathbf{W}^{in(v)} \\ \mathbf{W}^{out(v)} \end{bmatrix} = \begin{bmatrix} w_1^{in(v)} & \dots & w_i^{in(v)} & \dots & w_n^{in(v)} \\ w_1^{out(v)} & \dots & w_i^{out(v)} & \dots & w_n^{out(v)} \end{bmatrix}$ and $[\mathbf{W}^{(r)}] = \begin{bmatrix} \mathbf{W}^{in(r)} \\ \mathbf{W}^{out(r)} \end{bmatrix} = \begin{bmatrix} w_1^{in(r)} & \dots & w_i^{in(r)} & \dots & w_n^{in(r)} \\ w_1^{out(r)} & \dots & w_i^{out(r)} & \dots & w_n^{out(r)} \end{bmatrix}$.

Actually, when the topology of an LRS is determined, deciding how to design an appropriate set of parameters for each monitor is the core task of the supervisor synthesis. In other words,

$\{[W^{(v)}], M_0(v)\}$ needs to be calculated according to $\{[W^{(r)}], M_0(r)\}$ that is already known. A well-designed control path t_1vt_h , as a matter of fact, replaces the competition path t_1rt_h to be controlled in a WSDC. The quotas on tokens in r of upstream and downstream activity places are restricted by the designed numerical relationship among the calculated parameters.

First, parameters $\left\{ \begin{bmatrix} w^{in(v)} \\ w^{out(v)} \end{bmatrix}, M_0(v) \right\}$ of only one control path t_1vt_h is calculated. After the control path is imposed on a competition path t_1rt_h , two paths have the same upstream and downstream activity places, i.e., p_{up} and p_{down} . In order to make the control path satisfy Restriction 3.1, the number of tokens that is adequate for at least one operation of the downstream activity place, i.e., $M_0(r) \cdot \min \lambda^{r \rightarrow p_{down}}$ tokens, should be reserved in r . It means that the number of remaining tokens in r , i.e., $M_0(r) \cdot (1 - \min \lambda^{r \rightarrow p_{down}})$, can be prioritized and maximally required by the upstream activity place p_{up} . The maximal number of consecutive requirements of tokens in r by p_{up} is $\lfloor \frac{M_0(r) \cdot (1 - \min \lambda^{r \rightarrow p_{down}})}{w^{in(r)}} \rfloor$. Therefore, in order to ensure that the number of remaining tokens is still adequate for at least one operation of p_{down} , an inequality about parameters of the control place, $M_0(v) - w^{in(v)} \cdot \lfloor \frac{M_0(r) \cdot (1 - \min \lambda^{r \rightarrow p_{down}})}{w^{in(r)}} \rfloor \geq w^{out(v)}$, should hold as a key condition in the following mathematical programming model employed to solve all desired control parameters. The objective function $\min M_0(v) + w^{in(v)}$ is set to obtain the set of minimal control parameters to decrease control costs. In addition, both $M_0(v) \geq 1$ and $w^{in(v)} > w^{out(v)} \geq 1$ should hold at the same time according to Restriction 3.1. $M_0(r)$, $w^{in(r)}$, and $\min \lambda^{r \rightarrow p_{down}}$ are known conditions given a competition path t_1rt_h . Note that the complexity of solving the mathematical programming problem is NP-hard, although such problem with a small number of constraints and variables can be solved in very short time.

$$\begin{cases} \min M_0(v) + w^{in(v)} \\ s. t. \\ M_0(v) - w^{in(v)} \cdot \lfloor \frac{M_0(r) \cdot (1 - \min \lambda^{r \rightarrow p_{down}})}{w^{in(r)}} \rfloor \geq w^{out(v)} \\ M_0(v) \geq 1 \\ w^{in(v)} \geq w^{out(v)} + 1 \\ w^{out(v)} \geq 1 \end{cases}$$

For examples, the control path $t_2v_1t_7$ is imposed to control competition path $t_2p_9t_7$ in Figs. 5.8(a) and (b). $M_0(p_9) = 5$ and $w^{in(p_9)} = W(t_2, p_9) = w^{out(p_9)} = W(p_9, t_7) = 1$. The minimal non-zero

RU-ratio of p_9 with respect to its downstream activity place p_8 is $\min \lambda^{p_9 \rightarrow p_8} = 1/5$. Specifically, we have

$$\begin{cases} \min M_0(v_1) + w^{in(v_1)} \\ s. t. \\ M_0(v_1) - w^{in(v_1)} \cdot \lfloor \frac{5 \cdot (1 - 1/5)}{1} \rfloor \geq w^{out(v_1)} \\ M_0(v_1) \geq 1 \\ w^{in(v_1)} \geq w^{out(v_1)} + 1 \\ w^{out(v_1)} \geq 1 \end{cases}$$

Solving the above mathematical programming problem, we can obtain $M_0(v_1) = 9$, $w^{in(v_1)} = W(t_2, v_1) = 2$, and $w^{out(v_1)} = W(v_1, t_7) = 1$, i.e., $\left\{ \begin{bmatrix} 2 \\ 1 \end{bmatrix}, 9 \right\}$, the set of minimal control parameters, as shown in Fig.5.8(b). Under the supervision of monitor v_1 , it is determined that the upstream activity place p_2 's maximal quota is $4/5$ full capability of resource place p_9 , whereas the downstream one p_8 's minimal quota is $1/5$ full capability of p_9 . Note that the current admissible range of $\lambda^{p_9 \rightarrow p_2}$ is $[4/5, 0]^{p_9 \rightarrow p_2}$ in the controlled model instead of the original admissible range $[1, 0]^{p_9 \rightarrow p_2}$. The WS³PR net model in Fig. 5.8(a) has total 1500 reachable states including three dead ones. With the LRS in Fig. 5.8(b), the controlled model is live and has total 1190 states.

Next, when considering n control paths that are concerned with the same monitor v , the calculation is similar to solve the above mathematical programming problem. We use $\min M_0(v) + \sum_{i=1}^n w_i^{in(v)}$ as the objective function to obtain the values of all control parameters as small as possible. The restrictive inequalities for every control path are all listed in this mathematical programming problem. Note that the same restriction as condition (2) of Restriction 3.1 should also be considered, i.e., there is no such an n -dimensional row vector $\mathbf{B} = [b_1 \dots b_i \dots b_n]$ that $0 \leq (M_0(v) - \mathbf{B}[\mathbf{W}^{in(v)}]^T) < \min\{w_i^{out(v)}\}$, where $b_i \in \mathbb{N}$.

In fact, a designer can set different admissible ranges of RU-ratios of upstream and downstream activity places through adjusting $(1 - \min \lambda^{r \rightarrow p_{down}})$ in the inequality by using different ratios to replace the minimal non-zero RU-ratio $\min \lambda^{r \rightarrow p_{down}}$. This makes sense to implement a precise usage of resources, improve the robustness of an LRS-controlled system, or balance the workload. In other words, the resource usage ratio-enforcement can be achieved. Furthermore, when the initial marking and arc-weights of a plant model change, the topology of LRS remains unchanged,

and only new parameters are needed to respond to the new situations. Hence, the hard-wired jobs of controller re-design are avoided. For example, a resource in failure may lose a part of its capability. The proposed LRS can well deal with this kind of problems to keep this resource running in a system. In brief, the new supervisor proposed in this chapter achieves the goal of ratio-enforcement as well as liveness-enforcement.

5.3 Discussion

During the past two decades, a great number of liveness-enforcing methods were successively proposed and developed for FMS modeled with Petri nets. Most of them achieve their goals based on a significant structural object, i.e., siphons. The LRS presented in this chapter chooses an alternative to reach the liveness-enforcement. A special structure, WSDC, reflecting wait relations among resources has been fully explored in previous chapters, and laid a foundation for the realization of LRS. The structure of a plant model and its dynamic properties are considered together and a numerical relationship between initial markings and arc-weights is completely revealed to implement liveness-enforcement and ratio-enforcement of LRS.

In literature known to the authors, a relatively small amount of work focuses on ratio problems in FMS. In [36]-[32], the production ratio is considered in an FMS and enforced by adding a ratio controller for every pair of two different work processes. In [75], based on timed event graphs, modeling and cyclic scheduling problems for a production ratio-driven FMS are studied. In [47], an arc ratio between two transitions t_1 and t_2 , and a place p is defined as $\lfloor \frac{W(t_1, p)}{W(p, t_2)} \rfloor$ and used to develop a modular transformation method for automated manufacturing systems modeled by generalized Petri nets. Being different from them, the resource usage ratio is proposed and employed to probe into its impact on system liveness. The newly proposed supervisor combining liveness-enforcement with ratio-enforcement intends to realize a more precise segment usage of system resources.

In previous chapters, an intrinsically live structure is completely investigated in WS³PR. A WS³PR is live if every WSDC containing at least one competition path satisfies Restriction 3.1.

5.3. DISCUSSION

Inspired by this, a WS³PR can be controlled by inserting a control path satisfying Restricion 3.1, into a WSDC to virtually replace a competition path that does not satisfy it. One external monitor is added for each of such resource places. Since the connection of an LRS monitor is exactly the same as the resource place to be controlled, there is no new problematic structural objects (siphons or WSDCs) generated. Therefore, the size of an LRS is limited by the number of resource types, i.e., $|P_R|$, given a plant model. The control cost can be easily estimated before designing a specific LRS. In addition, the arc connections of a monitor are exactly the same with the resource place to be controlled. As a result, the topology of an LRS is also determined when a plant model is given. It makes this control method easy and clear enough to be understood and used by industrial practitioners. When a plant model is re-configured, i.e., the initial marking of resource places and arc-weights of competition paths are changed, the designed LRS can still work by adjusting its parameters only, i.e., the initial marking of monitors and arc-weights of control paths. It resembles the style of a programmable logic controller. The job of re-hard-wiring is avoided, and the changing control specification is responded by varying new control parameters.

The parameters of a control path include an initial marking of the monitor and the weights of its in-arc and out-arc. A special numerical relationship among them is found and used to cut quotas on the capability of the resource place for an upstream and an downstream activity place. It is not simply making a division of all tokens in the resource place for the upstream and downstream activity ones. In the example discussed above, the resource place p_9 in Fig. 5.8(a) has five tokens. The imposed LRS does not fix one token for the downstream activity place p_8 , and neither fix the remaining four for the upstream one p_2 . Actually, this supervisor imposes a limitation on the admissible range of the RU-ratio of upstream activity place p_2 with respect to resource place p_9 . It ensures that the upstream activity place cannot acquire all capability of the resource place, and the remaining tokens in the resource can be still adequate for an operation of the downstream one. The liveness-enforcement is accomplished by this ratio-enforcement that sacrifices the minimal part of the flexibility of a system. The admissible ranges of RU-ratios can be adjusted to gain a precise segment usage of the resources in a system or balance the workload. This is the very goal of LRS instead of obtaining the maximal number of reachable states, which makes more sense in theory.

The application scope of the proposed LRS covers a subclass of generalized Petri nets, WS³PR.

5.4. SUMMARY

The major obstacle to extend this kind of supervisors is an extended concept of WSDC in a more general subclass. Note that the existence of an LRS depends on the plant model. For example, if there is only one token in a resource place, a competition path containing it cannot be controlled by any LRS. In other words, the initial marking of a WS^3PR may affect the effectiveness of the proposed LRS method. The work in the chapter shows that if a feasible solution can be obtained from the formulated mathematical programming problem, the corresponding LRS exists, and otherwise it does not.

Note that the liveness-enforcement and resources usage ratio are the main focuses in this chapter. Since liveness is a behavioral property of Petri nets and the mechanism behind LRS that controls the maximal number of consecutive firings of transitions is also a limitation to system behavior, the RU-ratio is viewed as a behavioral indicator of a system from the perspective of logical causality. When the temporal parameter is introduced in the system, we need other methods to determine RU-ratios of the system to reach an optimal performance, such as throughput maximization and makespan minimization, which are very challenging problems.

5.4 Summary

A new liveness and resource usage ratio-enforcing supervisor is proposed in this chapter. By using the concept of resource usage ratios, the essence of an intrinsically live structure is analyzed and adopted to design an LRS, which makes a WS^3PR model live through controlling the maximal RU-ratios of upstream activity places with respect to corresponding resource places. An appropriate scheme of quotas on resource capabilities is implemented for the competitive operations by this supervisor. The connection of monitors is exactly the same as that of the corresponding resource places. The number of monitors is bounded by that of resource places in a plant model. Moreover, an LRS is highly reusable by adjusting control parameters. All the above features make it easy to be used by designers, researchers, and engineers. The future work includes choosing an optimal set of competition paths to be controlled, which yields as many reachable states of the controlled system as possible. In order to avoid the NP-hard problem in mathematical programming, another way to compute control parameters also needs to be considered by exploring the

5.4. SUMMARY

characteristic structure of WS^3PR . Extension to LRS applied to a more general class of Petri nets modeling more complex resource allocation systems remains open.

5.4. SUMMARY

Chapitre 6

Liveness-enforcement for WS^3PR Using a Divide-and-Conquer Paradigm

As a significant paradigm in computer science, the divide-and-conquer (D&C) technique is investigated for enforcing liveness of WS^3PR modeling FMS in this chapter. In the dividing stage, a primary separation and an advanced one of the whole structure of a plant net model are successively presented. After the decomposition and simplification, the real niduses of non-liveness can be precisely located in the whole structure. In the conquering stage, a kind of liveness and resource usage ratio-enforcing supervisors (LRSs) is adopted to conquer subnets without introducing new problematic siphons. By a combination of all subnets, a live net model is gained, in which all siphons are minimally controlled. The work in the chapter provides a deeper insight into structure theory and makes feasible the analysis and control of large-scale FMS modeled by generalized Petri nets. Several examples are used to illustrate the proposed method.

6.1 Introduction

An easy-to-understand example of the D&C method is given in [48], with which a post office routes mails : letters are sorted into separate bags for different geographical areas, each of these bags is itself sorted into batches for smaller sub-regions, and so on until they are delivered. In computer science, D&C is an important paradigm to design algorithms that iteratively divide one

problem into two or more subproblems or decrease one problem into a simple subproblem until the subproblems become simple enough to be completely handled. Then the solutions to the subproblems are combined to give a solution to the original problem. The D&C strategy is also used in the field of modeling and analyzing Petri nets [62].

Process mining that extracts a process model from the logs of a system is studied in [8] using the theory of regions with the D&C strategy. Hierarchical high level Petri nets and hierarchical reachability graph generation are explored in [5] and [6] using D&C approaches, respectively. In [20], a decomposition method is presented to automatically obtain sub-models of TimeNET that are later aggregated in an iterative way. The work in [45] proposes a formal verification methodology that smoothly integrates with the component-based system level using a D&C approach. The work in [46] points out that the modular verification based on the D&C principle is a promising natural approach to tackle the state explosion problem. The state space of a class of stochastic Petri nets is reduced in a D&C fashion in [71]. The well known siphon-trap property is re-considered in [74]. One of two new methods contributing to this issue has a D&C nature which builds upon the decomposition of a Petri net into open nets and projects information about siphons and traps onto the interfaces of the components. The work in [83] deals with supervisory control of discrete event systems modeled by interpreted Petri nets and presents a method to compute the system firing transition sequence in order to confine the system model into the specification behavior by using a D&C technique. It claims that the decentralized supervisory control is an effective method for large complex systems according to the D&C principle in [84]. Analysis of Petri nets using transitive matrix is implemented in [85] by proposing a D&C method under the condition of one-boundedness of Petri nets. The work in [86] reduces a Petri net representation to the desired level of details using abstraction of subnets to transitions, which can be used as a D&C approach to the analysis of Petri nets. In a summary, the D&C technique involved in the above studies is used to deal with either structure or state analysis of Petri net models.

In our previous work [62], a D&C strategy to deadlock prevention in FMS modeled by S^3PR is thoroughly elaborated. An S^3PR net is first divided into several kind of subnets, and then each of them is checked to determine whether it contributes to deadlocks. Then, sub-supervisors are designed for the subnets to be controlled by using elementary siphons. Finally, all subnets with

their external control are combined together to synthesize a controlled S^3PR net. In [27], it extends the strategy to WS^3PR , a more general subclass than S^3PR . The intrinsically live structures in subnets are identified to decide which subnet is live at the existing initial marking and hence no control needs to be imposed.

In this chapter, the D&C liveness-enforcing method is further investigated in WS^3PR . By fully utilizing the global and local intrinsically live structures, a WS^3PR is first considered under primary and advanced separations and then the remaining subnets are controlled by liveness and ratio-enforcing supervisors (LRS). The way of breaking the control problem into sub-problems and the application of LRS are the kernel of the work that provides a deeper insight into model structures and makes the analysis and control of complex WS^3PR feasible. The different control effects of LRS with different control parameters are also discussed in detail.

6.2 Dived-and-conquer strategy

This section presents a D&C strategy for liveness-enforcement of WS^3PR on the ground of our work in previous chapters. Besides that a primary partition separates a plant net model into some small but independent subnets, an advanced one is further carried out to reduce them and precisely locate the real niduses of non-liveness. Then, the LRS method is adopted to conquer every reduced small subnet resulting from the two dividing steps. Finally, a controlled WS^3PR is gained by the synchronous synthesis of all these controlled subnets.

The control mechanism of monitors in LRS is completely different from the traditional siphon-based ones. As shown in Fig. 6.1(a), a WS^3PR net is employed to illustrate the differences between them. (1) The structural objects tied with deadlock-freedom or liveness are different. Instead of a siphon $S = \{p_3, p_6, r_1, r_2\}$ shown in Fig. 6.1(c), a WSDC $c_W = t_2r_1t_5r_2t_2$ shown in Fig. 6.1(b) is first characterized when the LRS method is applied to control the plant model. (2) The control objects are different. The competition path $t_2r_1t_5$ is chosen to be controlled rather than the complementary set¹ of siphon S , i.e., $\{p_2, p_5\}$. (3) The control implementations are different. The LRS monitor v_1

¹The complementary set of a siphon includes holders of all resource places of the siphon but none in the siphon.

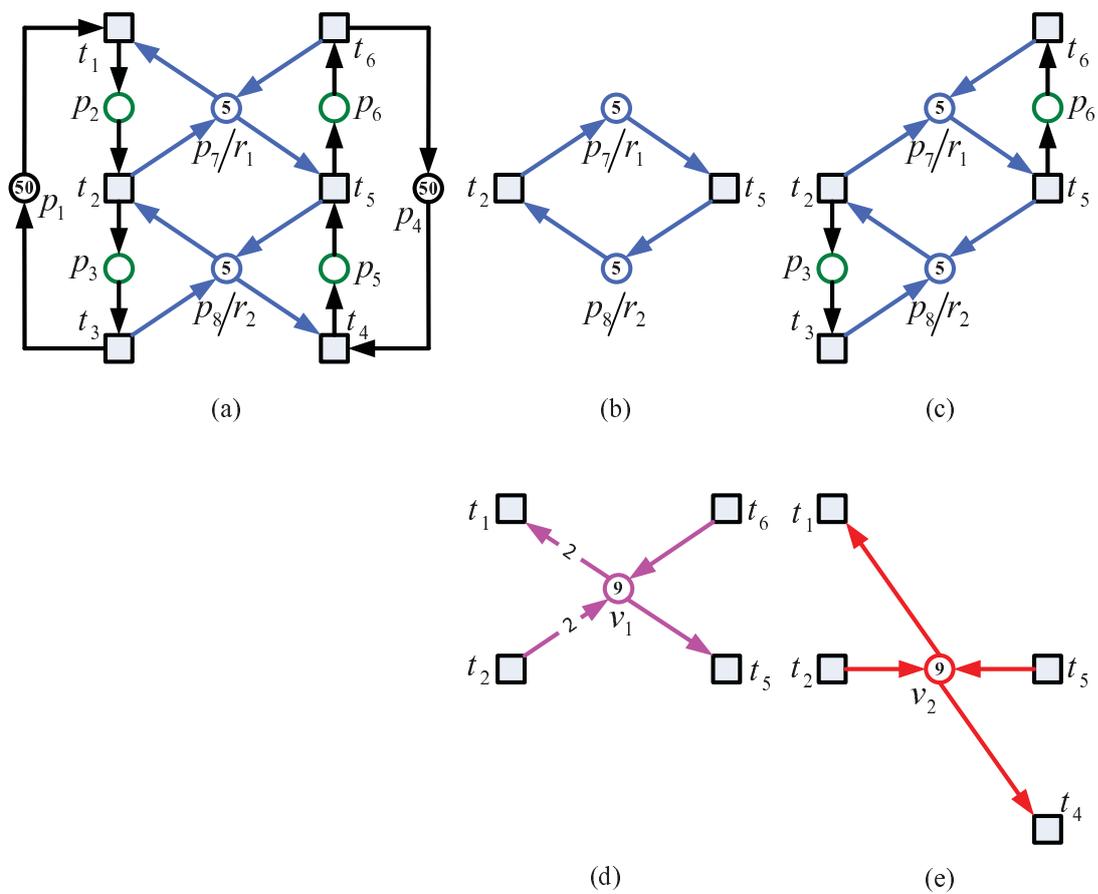


FIG. 6.1 – (a) A WS^3PR net model to be controlled , (b) a WSDC , (c) a subnet containing a strict minimal siphon, (d) an LRS monitor, and (e) a traditional siphon-based monitor.

together with all corresponding arcs shown in Fig. 6.1(d) forcibly takes over the resource allocation between the upstream activity place p_2 and the downstream one p_6 of the competition path $t_2r_1t_5$. Different from v_1 , monitor v_2 shown in Fig. 6.1(e) limits the total number of tokens held by the complementary set $\{p_2, p_5\}$ of a siphon to ensures enough ones remaining in $S = \{p_3, p_6, r_1, r_2\}$.

A monitor of LRS corresponds to a shared resource place to be controlled. The arc connection of the monitor is the same with that of the resource place. The control parameters, i.e., the initial number of tokens in the monitor and the weights of connecting arcs, can be obtained by solving a mathematical programming problem according to Restriction 3.1. For economy of space, please refer to Chapter 5 for LRS design in detail. It is adopted to control some subnets after a decomposition of the whole plant model.

6.2.1 Primary separation of toparchies

A primary separation to divide WS³PR into several components is first introduced, which is similar to the procedure in [62]. The terminologies and algorithms in [62] are partially adopted in this chapter. This part focuses on the differences from and the improvements to the studies in [27] and [62]. An illustrative example is employed to demonstrate the first step of the dividing stage for WS³PR.

As shown in Fig. 6.2, the net model is a WS³PR with process idle places $\{p_1, p_9, p_{12}, p_{16}\}$, activity places $\{p_2 - p_8, p_{10}, p_{11}, p_{13} - p_{15}, p_{17}, p_{18}\}$, and resource places $\{r_1 - r_7\}$. It has in total 3,090,672 reachable markings and 10 dead transitions among all 19 ones. Hence, it is deadlock-free but not live.

Before applying the existing method, a mistake in [62] must be pointed out and corrected. It claims that an S³PR N can be converted into a directed graph $G_N = (V, E)$ by the following steps : 1) $V = P_R$ and 2) given $r, r' \in P_R$, there is an edge in E from r to r' if $r^\bullet \cap \bullet r' \neq \emptyset$. As a result, finding a resource circuit (equivalent to the concept of simple CW in the chapter) in N is equivalent to finding a cycle in G_N . Since the directed graph G_N defined in [62] does not include transitions, the resource circuits with the same resource places but different transitions cannot be distinguished from G_N . For example, $c_{W1} = t_8r_6t_{18}r_7t_8$ and $c_{W2} = t_{11}r_6t_{18}r_7t_{11}$ in Fig. 6.2 fall

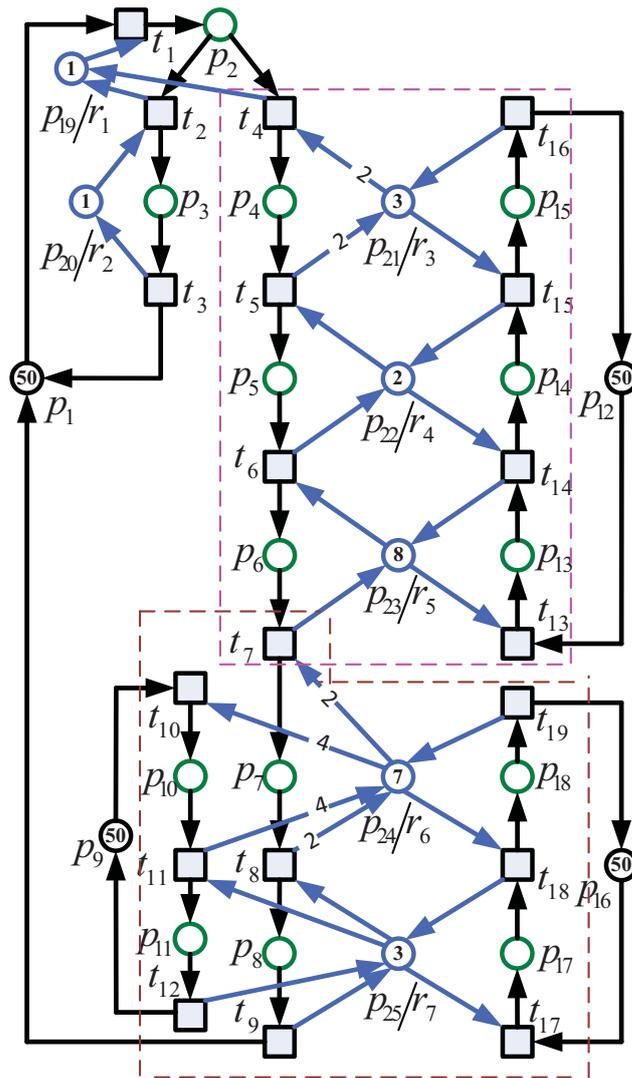


FIG. 6.2 – A WS³PR net model.

in the aforementioned ambiguous situation. To avoid this, G_R , the weighted digraph of resource places is used to identify all WSDC (i.e., all resource circuits).

=====

Algorithm 6.1

Partition computation of P_R .

Input : A weighted digraph of resource places G_R derived from a WS³PR $N = (P_0 \cup P_A \cup P_R, T, F, W)$.

Output : $P_R^1, P_R^2, \dots, P_R^m$, and P_R^F .

The procedures are the same as Algorithm 2 in [62].

=====

Since the topological structures of S³PR and WS³PR are the same, Algorithm 2 in [62] can be also used to split the set of resource places P_R in a WS³PR into a number of subsets $P_R^1, P_R^2, \dots, P_R^m$, and P_R^F such that $P_R = \cup_{i=1}^m P_R^i \cup P_R^F$ and $\forall i, j \in \mathbb{N}_m, i \neq j, P_R^i \cap P_R^j \cap P_R^F = \emptyset$. Actually, P_R^i expresses a set of resource places that forms a strongly connected component in G_R of N , whereas P_R^F is a set of resources places and does not exist in G_R , i.e., each of them is not included by any simple CW. After the application of Algorithm 6.1, the resource places in Fig. 6.2 are partitioned into $P_R^1 = \{r_3 - r_5\}$, $P_R^2 = \{r_6, r_7\}$, and $P_R^F = \{r_1, r_2\}$. In fact, the dividing work in [62] is just based on the separation of strongly connected components that contain different resource places. However, it is only the primary step of the dividing stage in this chapter.

A marked subnet (N^i, M_0^i) derived from $P^i \cup T^i$ in a WS³PR (N, M_0) with $N = (P_0 \cup P_A \cup P_R, T, F, W)$ is called a toparchy derived from P_R^i , where $P^i = P_R^i \cup \{p \in P | p \in H(r), r \in P_R^i\}$ and $T^i = \cup_{p \in P^i} (\bullet p \cup p \bullet)$. Its weighted digraph of resource places of a toparchy is denoted by G_R^i . The subnet (N^F, M_0^F) derived from $P^F \cup T^F$ in a WS³PR is called an autonomous subnet derived from P_R^F , where $P^F = P_R^F \cup \{p \in P | p \in H(r), r \in P_R^F\}$ and $T^F = \cup_{p \in P^F} (\bullet p \cup p \bullet)$. Since the resource places in P_R^F do not belong to any strongly connected component, the autonomous subnet has no deadlock ; therefore, only the toparchies need to be carefully considered.

Before the conquering stage, each toparchy of the original plant net model is checked to determine whether its idle-augmented net, i.e., a synthesis of the toparchy and corresponding process idle places, is live. A live (resp. non-live) toparchy is said to be subordinate (resp. dominant). In [62], a decision method is reported by considering the token counts in resource and process idle places. In [27], a new method that does not limit the tokens counts in process idle places, which are less than those in resource places, is presented by considering the relation between structures and initial numbers of tokens in the resource places. If all WSDCs in a toparchy satisfy Restriction 3.1, the idle-augmented net of this toparchy is live. For example, both WSDCs $c_{W1} = t_8r_6t_{18}r_7t_8$ and $c_{W2} = t_{11}r_6t_{18}r_7t_{11}$ in the toparchy derived from $P_R^2 = \{r_6, r_7\}$ satisfy Restriction 3.1 in Fig. 6.2. Hence, this toparchy is subordinate and no control needs to be imposed. Conversely, in the other toparchy derived from $P_R^1 = \{r_3 - r_5\}$, $c_{W3} = \{t_5r_3t_{15}r_4t_5\}$ satisfies Restriction 3.1, whereas $c_{W4} = \{t_6r_4t_{14}r_5t_6\}$ does not. Hence, this toparchy is a dominant one.

Note that the existence of an autonomous subnet derived from P_R^F is not necessary for WS^3PR . Furthermore, the high sharing of resources may connect all resource places in P_R into one unique but complex strongly connected component of G_R . In addition, the requirement of a global intrinsically live structure of a toparchy may be too strong to fulfill. Therefore, the D&C strategy proposed in [27] [62] cannot effectively work on such WS^3PR . A further decomposition of a toparchy is investigated next, which was not reported previously.

6.2.2 Advanced separation of toparchies

In Chapter 4, the ILSs are used to identify and remove redundant elementary siphons (a set of siphons chosen from all strict minimal ones) to be controlled. Inspired by this, the basic idea is directly peel off local intrinsically live structures of a toparchy obtained after a primary separation. The real nidus of non-liveness in a toparchy can be precisely located for conquering by LRS.

Since the target objects of the LRS control method are resource places and competition paths, we use the weighted digraph of resource places G_R^i of a toparchy N^i derived from P_R^i to carry out the second step of the dividing stage for simplicity. The advanced separation of a toparchy consists of two parts. In the first part, every resource place in G_N^i of the toparchy N^i is checked to determine whether it can be taken away. In the second part, every WSDC is checked to determine whether

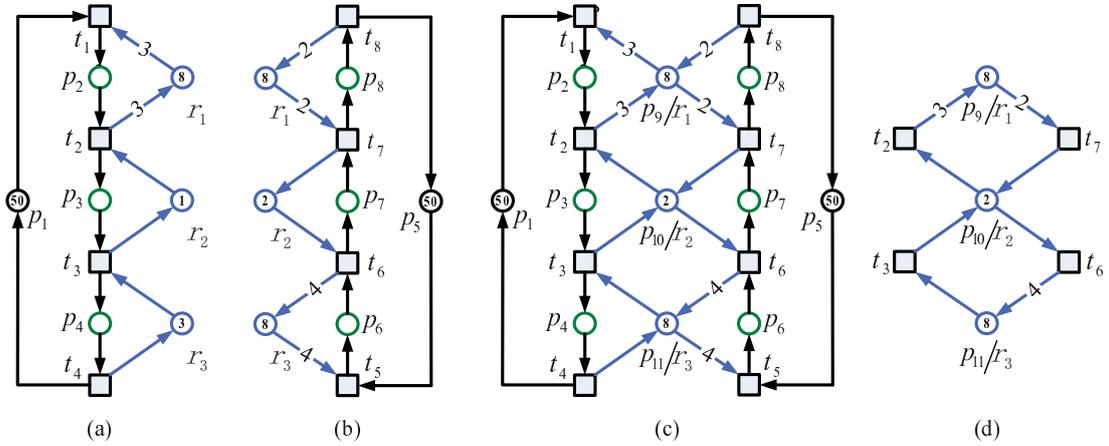


FIG. 6.3 – (a) A generalized Petri net (N_1, M_{01}) , (b) another generalized Petri net (N_2, M_{02}) , (c) a composed net (N, M_0) , and (d) a subnet of (N, M_0) .

its several nodes and arcs can be removed to further reduce the local structure to be controlled. Algorithms are developed to implement the advanced separation.

=====

Algorithm 6.2

Reduction of the weighted digraph of resource places of a toparchy by checking every resource places.

Input : G_R^i and $M_0(P_R^i)$.

Output : A reduced weighted digraph of resource places G_R^{i*} .

Begin

$k := 0$;

while ($k \leq |P_R^i|$)

 { $k := k + 1$;

if (all competition paths associated with r_k satisfy Restriction 3.1)

 { **if** (no paired arcs with r_k exist in G_R^i)

 { Remove r_k and all arcs connecting to it from G_R^i ;

 Remove all nodes without input/output arcs and arcs connecting

6.2. DIVED-AND-CONQUER STRATEGY

to them from G_R^i until no such nodes exist in G_R^i ;

$$P_R^i := P_R^i - \{r_k\};$$

}

}

}

$$G_R^{i*} := G_R^i; P_R^{i*} := P_R^i;$$

Output G_R^{i*} and P_R^{i*} ;

End.

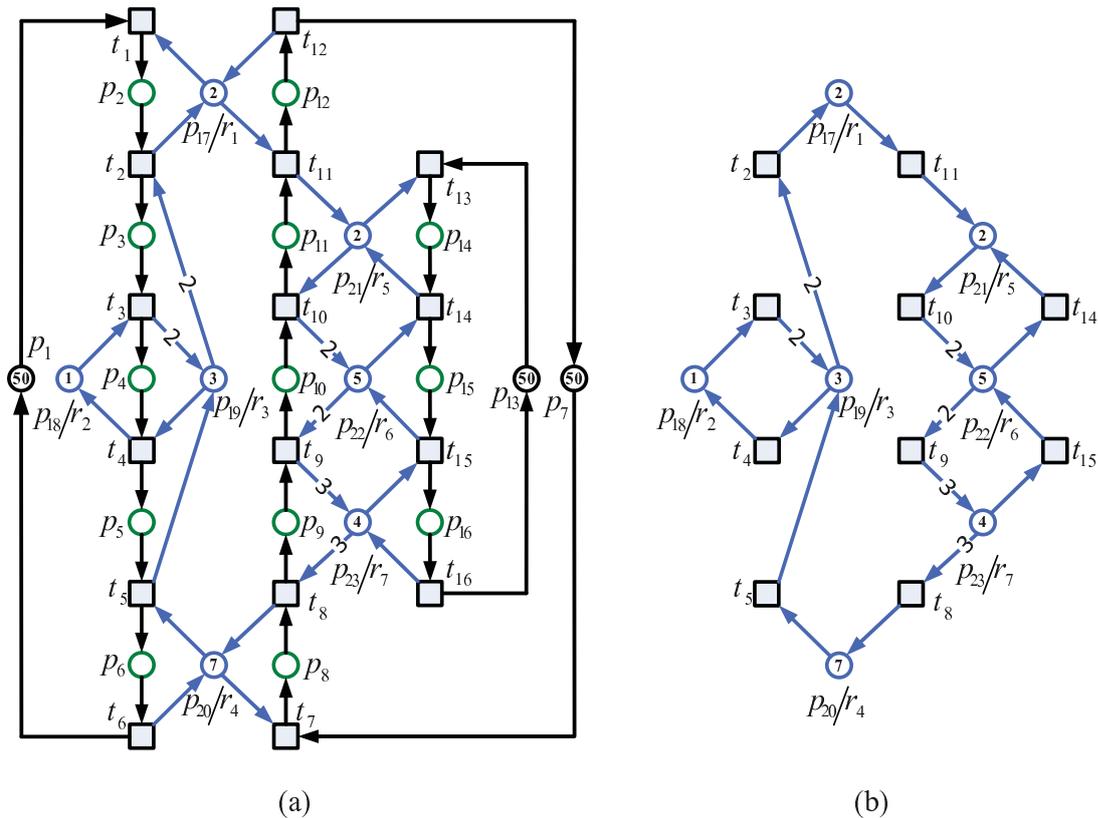


FIG. 6.4 – (a) A WS³PR net with only one toparchy and (b) its weighted digraph of resource places.

For the net shown in Fig. 6.3(d), its G_R in Fig. 6.3(c) is the unique strongly connected com-

ponent. In other words, there is no autonomous subnet and only one toparchy derived from $P_R = \{r_1 - r_3\}$ in the plant net model. We check every resource place first and find that r_1 satisfies Restriction 3.1 and no paired arcs of it exist in the subnet. Hence, r_1 and its connecting arcs can be removed from the subnet. In addition, the resulting transitions without any input or output arc are also removed from the subnet. It means that every WSDC contains r_1 does not contribute to non-liveness ; therefore, no control needs to be imposed on this part of structures. Then, the other resource places are checked and they do not satisfy Restriction 3.1 and are kept in the subnet. Consequently, the remaining part after the advanced separation is just one WSDC $t_3r_2t_6r_3t_3$. There are two competition paths $t_3r_2t_6$ and $t_6r_3t_3$ existing in it, and each of them can be chosen to add an LRS monitor

The above example explains the advanced separation of a toparchy that involves only the first part. Without loss of generality, the situation that not all competition paths of resource place r may satisfy Restriction 3.1 must be considered. Algorithm 6.3 is developed to deal with this situation.

=====

Algorithm 6.3

Reduction of the toparchy by checking every WSDC.

Input : G_R^{i*} , all m WSDC contained in G_R^{i*} , and $M_0(P_R^{i*})$.

Output : A further reduced weighted digraph of resource place $G_R^{i\diamond}$.

Begin

$k := 0$;

while ($k \leq m$)

 { $k := k + 1$;

if (c_{Wk} satisfies Restriction 3.1)

 { Remove its nodes and corresponding arcs that do not belong to any other

 WSDC from G_R^{i*} ;

 Remove all nodes without input/output arcs and arcs connecting to them

 until no such nodes exist in G_R^{i*} ;

```

    }
  }
   $G_R^{i\diamond} := G_R^{i*}$ ;
  Output  $G_R^{i\diamond}$ ;
  End.

```

=====

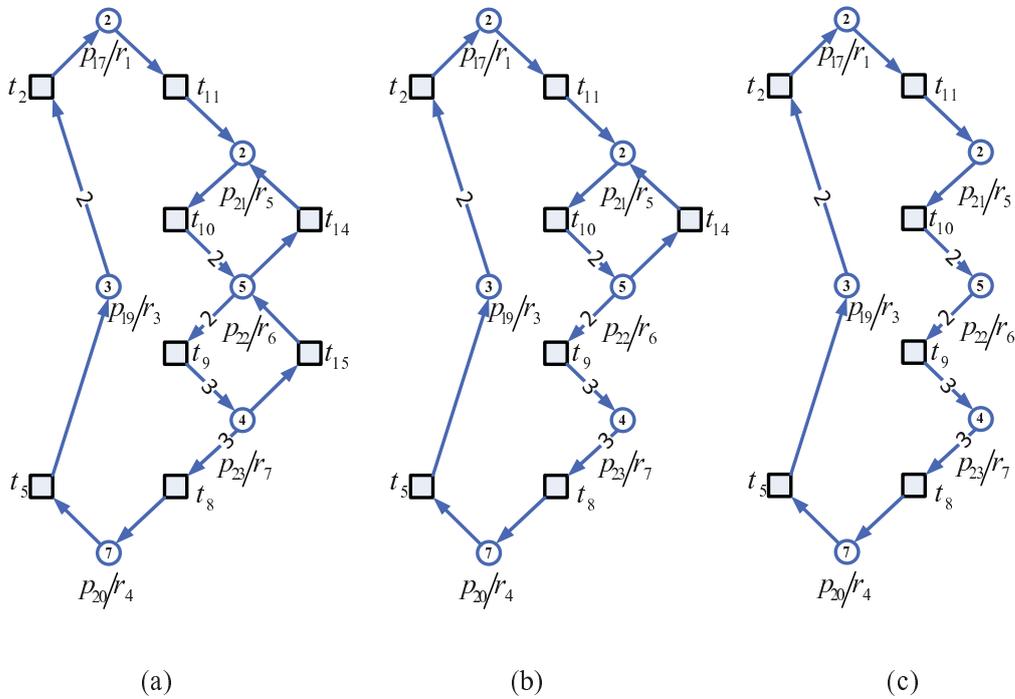


FIG. 6.5 – (a) A subnet after removing nodes and arcs of WSDC $t_3r_3t_4r_2t_3$ that do not belong to any other WSDC, (b) after removing nodes and arcs of WSDC $t_9r_7t_{15}r_6t_9$ that do not belong to any others, and (c) the final remaining subnet to be controlled after removing nodes and arcs of WSDC $t_{10}r_6t_{14}r_5t_{10}$ that do not belong to any others..

A WS³PR plant model and its G_R are shown in Fig. 6.4. There is only one strongly connected component existing in G_R . Hence, only one toparchy is derived from all resource places $P_R = \{r_1 - r_7\}$. By Algorithm 6.2, no resource place can be removed from G_R . Note that, although all competition paths with r_7 (actually, only $t_9r_7t_{15}$) satisfy Restriction 3.1, it cannot be removed since the existence of the paired arcs (r_7, t_8) and (t_9, r_7) . It means that r_7 may be contained by a WSDC

that does not contain r_7 's any competition path satisfying Restriction 3.1. However, the subnet $G_R^{i*} = G_R$ can be further reduced by Algorithm 6.3.

The aim of the advanced separation of a toparchy is to precisely locate the WSDC to be controlled. Fig. 6.4(b) and Fig. 6.5 are used to show the application of Algorithm 6.3. First, WSDC $t_3r_3t_4r_2t_3$ is checked to determine whether it contains a competition path satisfying Restriction 3.1. Since $t_3r_3t_4r_2t_3$ does so, it does not contribute to non-liveness. Resource place r_2 , t_3 and t_4 , and their corresponding arcs are not contained by any other WSDC and hence removed from G_R^{i*} . It means that we do not need to design any control for this part of structures at the current markings. Second, WSDC $t_9r_7t_{15}r_6t_9$ and $t_{10}r_6t_{14}r_5t_{10}$ are subsequently checked and those kinds of nodes and arcs are removed from G_R^{i*} . Finally, $t_2r_1t_{11}r_5t_{10}r_6t_9r_7t_8r_4t_5r_3t_2$ shown in Fig. 6.5(c) is checked. It does not satisfy Restriction 3.1 and no nodes and arcs can be removed. Since all WSDC have been checked by Algorithm 6.3, the final remaining component as shown in Fig. 6.5(c) is the structure of the original toparchy to be controlled.

Two parts of the advanced separation are separately illustrated with two examples, respectively. The remaining component of a weighted digraph of resource places of a toparchy is the very object to be controlled by applying LRS.

6.2.3 Toparch and controlled WS³PR synthesis

After the primary and advanced separations, the problem is converted into sub-problems to control several weighted digraphs of resource places derived and reduced from toparchies. A toparch, i.e., a sub-supervisor, is designed for each toparchy with LRS. As discussed in Section 3.2, the control objects of LRS are resource places and their competition paths and no new problematic siphon will be generated after adding LRS monitors. Because of these characteristics, an LRS monitor is particularly suitable for local controls.

The plant net model shown in Fig. 6.2 has totally 3,090,672 states and no dead one among them. It is deadlock-free but not live. The remaining subnet after primary and advanced separations is just WSDC $t_6r_4t_{14}r_5t_6$. The resource place r_5 and its competition path $t_{14}r_5t_6$ are chosen to be controlled by LRS. A control path $t_{14}v_1t_6$ shown in Fig. 6.6(a) is imposed on WSDC $t_6r_4t_{14}r_5t_6$,

i.e., G_R^{i**} . Its paired arcs (t_7, v_1) and (v_1, t_{13}) are supplemented in the toparchy. The controlled system is live and has 3,021,584 states. About 97.76% (3,021,584/3,090,672) of all reachable states are preserved under the control of LRS monitor v_1 shown in Fig. 6.6(a).

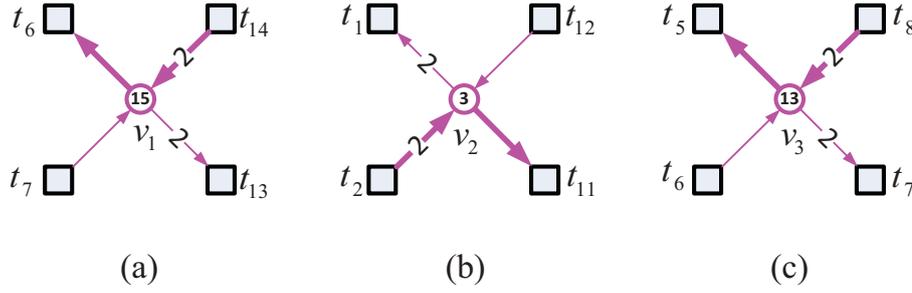


FIG. 6.6 – (a) An LRS monitor v_1 for r_5 shown in Fig. 6.2, (b) an LRS monitor v_2 for r_1 shown in Fig. 6.5(c), and (c) an alternative LRS monitor v_3 for r_4 shown in Fig. 6.5(c).

As aforementioned, a WSDC can be controlled by imposing a control path for only one of its competition paths. When more than one competition path exists in WSDC, in order to obtain more reachable states of the controlled system, choosing which one, actually, which resource place, needs a careful consideration. The optimal choice is still an open question, however. A guiding principle is given next. In order to achieve as many reachable states as possible, among all resource places in WSDC, the one with the largest value of $\frac{M_0(r)}{W^{out}(r)}$ should be chosen to be controlled by LRS. Actually, this is determined by LRS' own characteristics to limit the flexibility of plant systems.

For example, the plant net model shown in Fig. 6.4(a) has totally 933,112 reachable states and two dead ones among them. The controlled net under LRS monitor v_2 shown in Fig. 6.6(b) has 777,592 reachable states (83.33% of all reachable states of the original plant model), while it has 907,192 reachable states (97.22% of all reachable states of the original plant model) under v_3 shown in Fig. 6.6(c).

Even for the same LRS monitor, different control effects are expected by setting different control parameters that depend on different control specifications. The LRS monitor shown in Fig. 6.6(c) allows the maximal admissible range of RU-ratio $\lambda^{r_4 \rightarrow p_8}$ on the premise that the liveness is guaranteed. The maximal admissible range of RU-ratio usually implies the most reachable states

6.2. DIVED-AND-CONQUER STRATEGY

TABLE 6.1 – Different control effects of the LRS monitor v_3 with different control parameters for WS³PR shown in Fig. 6.4(a).

admissible range	$M_0(v_3)$	$w^{in(v_3)}$	$w^{out(v_3)}$	all reachable states	percentage	live
$[1, 0]^{r_4 \rightarrow p_8}$	/	/	/	933, 112	100%	No
$[6/7, 0]^{r_4 \rightarrow p_8}$	13	2	1	907, 192	97.22%	Yes
$[5/7, 0]^{r_4 \rightarrow p_8}$	17	3	1	855, 352	91.67%	Yes
$[4/7, 0]^{r_4 \rightarrow p_8}$	19	4	1	777, 592	83.33%	Yes
$[3/7, 0]^{r_4 \rightarrow p_8}$	19	5	1	673, 912	72.22%	Yes
$[2/7, 0]^{r_4 \rightarrow p_8}$	17	6	1	544, 312	58.33%	Yes
$[1/7, 0]^{r_4 \rightarrow p_8}$	13	7	1	388, 792	41.47%	Yes

of an LRS-controlled system, while a medium one usually provides a balanced workload at the cost of less reachable states. As shown in Table 7.1, the smaller admissible range of RU-ratio $\lambda^{r_4 \rightarrow p_8}$, the less reachable states of the LRS-controlled system. Note that the control parameters of LRS monitor v_3 , i.e., $M_0(v_3)$, $w^{in(v_3)}$, and $w^{out(v_3)}$, are calculated according to different control specifications, i.e., the different admissible range of RU-ratio $\lambda^{r_4 \rightarrow p_8}$ of the competition path $t_8 r_4 t_5$ to be controlled.

Definition 6.1 [62] Let (N_1, M_{01}) and (N_2, M_{02}) be two nets with $N_i = (P_i, T_i, F_i, W_i)$, $i = 1, 2$, satisfying $P_1 \cup P_2 = \emptyset$. (N, M_0) with $N = (P, T, F, W)$ is said to be a synchronous synthesis net resulting from the merge of (N_1, M_1) and (N_2, M_2) , such that (1) $P = P_1 \cup P_2$; (2) $T = T_1 \cup T_2$; (3) $F = F_1 \cup F_2$; (4) $W(f) = W_i(f)$ if $f \in F_i$, $i = 1, 2$; and (5) $M_0(p) = M_{0i}(p)$ if $p \in P_i$, $i = 1, 2$.

Similar to Definition 6.1, a re-combination is implemented after the application of the proposed D&C method to a WS³PR plant net model. A controlled WS³PR is obtained through the synchronous synthesis of all subnets including the autonomous one, the subordinate toparchies and the controlled dominant ones. As shown in Fig. 6.7, a schematic diagram depicts the framework of the D&C liveness-enforcing strategy for WS³PR.

6.3 Discussion

In [62], S^3PR is divided and conquered mainly according to strongly connected components of resource places. Later in [27], this D&C strategy is applied to a more general subclass WS^3PR and a class of intrinsically live structures is investigated to determine whether a toparchy is subordinate or dominant. In both, elementary siphons and other methods are employed to control the dominant toparchies. The work in previous chapters lays a foundation for the proposed one in the chapter.

The ILSs and derived LRS control method successively investigated in previous chapters provide a new perspective of breaking a control problem into a series of subproblems. The whole structure of a WS^3PR net model is first partitioned into several subnets that are relatively independent of each other in terms of circular waits of resources. Then, a subnet can be further reduced by a full consideration of local intrinsically live structures of the subnet. Hence, the real nidus of non-liveness of a subnet can be precisely located to be controlled by LRS monitors.

LRS is a new type of supervisors that achieve the goals of liveness- and ratio-enforcement of WS^3PR at the same time. The control objects of LRS are resource places and their competition paths that represent competition relations for resources among multiple operations. Compared with siphons, the number of structural objects to be controlled by LRS are much smaller. Due to the characteristics of LRS, there is no coupling among monitors imposed on different toparchies. Furthermore, under the supervision of LRS, all strict minimal siphons of a WS^3PR are minimally controlled and no new problematic siphon is generated. For example, in Fig. 6.4(a), it has total 6 strict minimal siphons, $S_1 = \{p_{11}, p_{16}, r_5, r_6, r_7\}$, $S_2 = \{p_{11}, p_{15}, r_5, r_6\}$, $S_3 = \{p_{10}, p_{16}, r_6, r_7\}$, $S_4 = \{p_6, p_{12}, p_{16}, r_1, r_2, r_3, r_4, r_5, r_6, r_7\}$, $S_5 = \{p_3, p_6, p_{12}, p_{16}, r_1, r_3, r_4, r_5, r_6, r_7\}$, and $S_6 = \{p_5, r_2, r_3\}$. The controlled model with an LRS monitor v_3 shown in Fig. 6.6(c) has 8 strict minimal siphons. Besides the above 6 ones, two new ones are generated as the result of the introduction of v_3 as a virtual resource place actually manipulating the resource allocation of r_4 , which are $S_7 = \{p_6, p_{12}, p_{16}, r_1, r_2, r_3, r_5, r_6, r_7, v_3\}$ and $S_8 = \{p_3, p_6, p_{12}, p_{16}, r_1, r_3, r_5, r_6, r_7, v_3\}$. In the controlled system, all 8 strict minimal siphons are min-marked at any reachable states, i.e., they are minimally controlled.

The reduced weighted digraph of resource places $G_R^{i\diamond}$ of a toparchy N^i derived from P_R^i is also

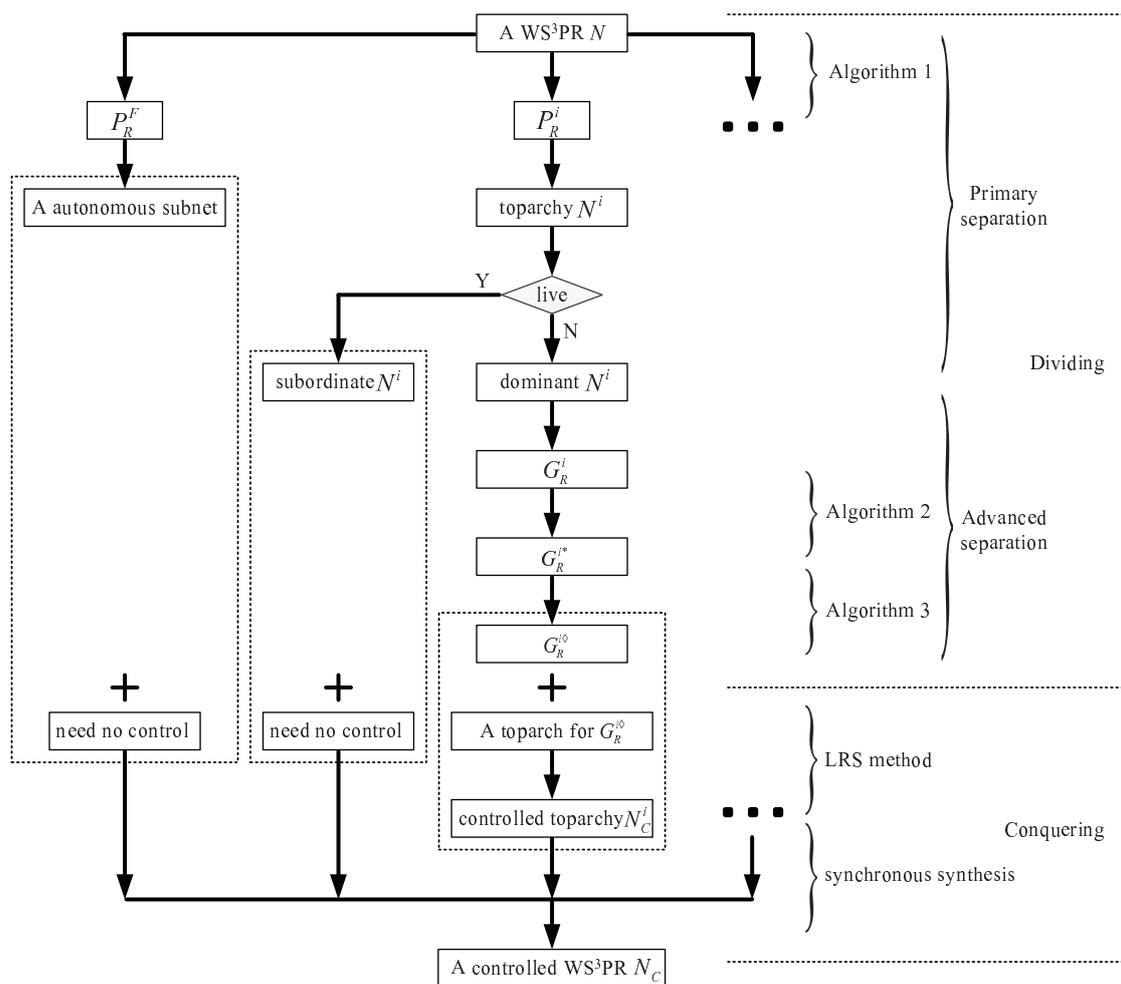


FIG. 6.7 – A schematic diagram of the D&C liveness-enforcing method for WS³PR.

6.3. DISCUSSION

a strongly connected component containing several WSDC. Hence, a priority for controlling a competition path that is included by as many WSDCs as possible should be established. It results in a small number of LRS monitors for a toparchy. As for a supremum of the number of LRS monitors of the D&C strategy for WS³PR, the following discussion can be given. Without loss of generality, suppose that a WS³PR net model with n resource places can be completely divided into m dominant toparchies to be controlled, and the numbers of resource places of toparchies are k_1, k_2, \dots , and k_m with $n = k_1 + k_2 + \dots + k_m$. Since a WSDC needs only one LRS monitor and due to the resource usage style of WS³PR, at most $k_i - 1$ LRS monitors are needed to control the i -th toparchy, $i \in \{1, 2, \dots, m\}$. Therefore, the total number of all involved LRS monitors for the WS³PR is less than or equal to $k_1 - 1 + k_2 - 1 + \dots + k_m - 1 = n - m$.

The reason that abundant initial markings are adopted in the work instead of acceptable ones is to ensure that there always exists at least one feasible LRS monitors for a WSDC. Otherwise, if there is only one token initially allocated in every resource place of a WSDC, it is not possible to design an LRS monitor to control it. However, the abundant initial markings do not guarantee that every resource place with a competition path in a particular WSDC can be controlled by an LRS monitor. It depends on the situation. For example, no feasible LRS monitor can be designed for r_3 and its competition path $t_5r_3t_2$ shown in Fig. 6.5(c). It can be explained by contradiction. Let v be an LRS monitor for r_3 . It demands $W(t_5, v) > W(v, t_2)$ to control the competition path $t_5r_3t_2$ and $W(t_3, v) > W(v, t_4)$ to reflect the competition path $t_3r_3t_4$ already satisfying Restriction 3.1. Since (t_3, v) and (v, t_2) and (t_5, v) and (v, t_4) are both paired arcs, $W(t_3, v) = W(v, t_2)$ and $W(t_5, v) = W(v, t_4)$ hold. Therefore, no feasible LRS monitor can be designed for r_3 . Note that when applying the proposed D&C method to WS³PR with acceptable but not abundant initial markings, the primary and advanced separations can be still used to partition the whole structure and reduce resulting subnets, while the LRS control method should be replaced by other siphons-based ones.

6.4 Summary

A D&C liveness-enforcing method for WS^3PR is proposed in this chapter. According to the strongly connected components of the resource places, a WS^3PR net can be primarily separated into several subnets, and then each of them can be further reduced with algorithms based on a full consideration of the intrinsically live structures. The genuine structures contributing to non-liveness can be precisely located for designing required monitors. The controlled subnets are re-combined together to solve the original control problem of the whole WS^3PR net model. All siphons in the controlled WS^3PR net are minimally controlled and the control effects resulting from different control parameters are discussed. The future work is twofold. The concept of WSDCs needs to be extended to identify more general intrinsically live structures in generalized Petri nets. The other worthy direction is the application of the proposed method to more general subclasses of Petri nets.

6.4. SUMMARY

Chapitre 7

Iterative Liveness-enforcement for WS^3PR

This chapter presents an iterative liveness-enforcing method for WS^3PR , which can model flexible manufacturing systems. The proposed method checks the liveness of net models using mixed integer programming and controls the token allocations of resource places instead of siphons using a liveness and resource usage ratio-enforcing supervisor. The enumeration of a kind of special structures, which is required in the work in previous chapters, is avoided and the number of iterations is bounded by the number of shared resource places in the net model. All strict minimal siphons in the controlled systems are minimally controlled.

7.1 Introduction

In the work [16], monitors are designed for siphons that are recognized as a significant structural object closely related with the liveness of S^3PR . Much work [58] focuses on designing siphon-oriented monitors with various strategies. However, the enumeration of siphons is necessary in most work. The number of all siphons is exponential with the size of the Petri net plant model. To avoid their enumeration, mixed integer programming (MIP) is proposed in [14] for augmented marked graph and developed in [10] [34] [37] [73] [121] to generate iterative siphon-control methods. A maximal or minimal insufficiently marked siphon is first found out by MIP and then

a monitor is designed to control the minimal one directly obtained by MIP or derived from the maximal one. These methods are experimentally proved to be computationally efficient.

In this chapter, MIP and LRS are combined together to yield an iterative liveness-enforcing method for WS³PR. The application of LRS no longer depends on an enumeration of WSDC but shared resource places in maximal insufficiently marked siphons found via MIP. The number of iterations can thus be limited. All strict minimal siphons in the controlled net model are minimally controlled. The work in the chapter lays a foundation for a direct application of LRS without enumeration of WSDC in Petri nets.

7.2 Liveness and ratio-enforcing supervisor

Inspired by the competition path satisfying Restriction 3.1, similar structural objects with proper initial token assignment can be designed to manipulate the token allocations in resource places of WSDC. A liveness and ratio-enforcing supervisor (LRS) achieves the liveness-enforcement through enforcing the resource usage ratios.

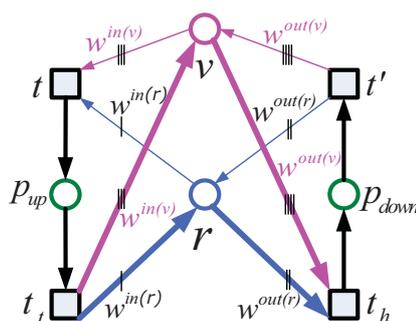


FIG. 7.1 – A competition path and its imposed control one.

An LRS monitor consists of a control path and its corresponding paired arcs. As shown in Fig. 7.1, $t_t r t_h$ is a competition path to be controlled. The RU-ratios of p_{up} and p_{down} with respect to r are $\lambda^{r \rightarrow p_{up}}$ and $\lambda^{r \rightarrow p_{down}}$, respectively. The control path $t_r v t'$ is used to ensure that $\min \lambda^{r \rightarrow p_{down}}$ can be reached. $M_0(v)$, $w^{in(v)}$, and $w^{out(v)}$ are called the control parameters of an LRS monitor. It

can be designed by solving the following optimization problem.

=====

Problem 1 :

$$\min M_0(v) + \sum w_i^{in(v)}$$

s. t.

$$M_0(v) - w_i^{in(v)} \cdot \lfloor \frac{M_0(r) \cdot (1 - \min \lambda^{r \rightarrow P_{down}})}{w_i^{in(r)}} \rfloor \geq w_i^{out(v)} ;$$

$$M_0(v) \geq 1 ;$$

$$w_i^{in(v)} \geq w_i^{out(v)} + 1 ;$$

$$w_i^{out(v)} \geq 1 ; \text{ and}$$

Condition (2) of Restriction 3.1 should also be satisfied, i.e., there is no such an n -dimension row vector $\mathbf{A} = [a_1 \dots a_i \dots a_n]$ that $0 \leq (M_0(v) - \mathbf{A}[\mathbf{W}^{in(v)}]^T) < \min\{w_i^{out(v)}\}$, where $i \in \mathbb{N}_n$ and $a_i \in \mathbb{N}$.

=====

Note that an LRS monitor v is corresponding to a resource place r . Its connection with transitions is exactly the same as r 's. The control parameters, i.e., $M_0(v)$, $w_i^{in(v)}$, and $w_i^{out(v)}$ are calculated to manipulate the token allocations of r and all its competition paths. The monitor v actually replaces the resource place r in WSDC and virtually acts a role to make all WSDC containing r satisfying Restriction 3.1.

7.3 MIP and LRS

In previous chapters, it is necessary to enumerate WSDCs in a WS³PR. From the graph theory, all simple (or called elementary) circuits of a directed graph (i.e., the weighted digraph of resource places in the chapter) can be found with time bounded by $O((n + e)(c + 1))$ and space bounded by $O(n + e)$, where there are n nodes, e arcs, and c simple circuits [43]. It is time-consuming when we consider large scale WS³PR net models with complex resource usages.

The MIP technology is extensively used for control design while avoiding the enumeration of

all siphons in a net. Although the computational complexity of MIP is NP-hard, it turns out to be computationally efficient when dealing with almost all plant net models. Therefore, an intuitive idea is that combining MIP with LRS monitors to yield an iterative liveness-enforcing method. The MIP in [121] is adopted to find a maximal insufficiently marked siphon (MIMS) if existing. Please refer to [121] for more details.

=====

Problem 2 :

$$G^{MIP} = \min \sum_{p \in P} v_p$$

s. t.

$$z_t \geq \sum_{p \in \bullet t} v_p - |\bullet t| + 1, t \in T ;$$

$$v_p \geq z_t, \forall (t, p) \in F ;$$

$$v_p, z_t \in \{0, 1\} ;$$

$$v_p \geq (M(p) - W(p, t) + 1)/B(p), \forall p \in P, \forall W(p, t) > 0 ;$$

$$M = M_0 + [N]Y, M \geq 0, Y \geq 0 ;$$

=====

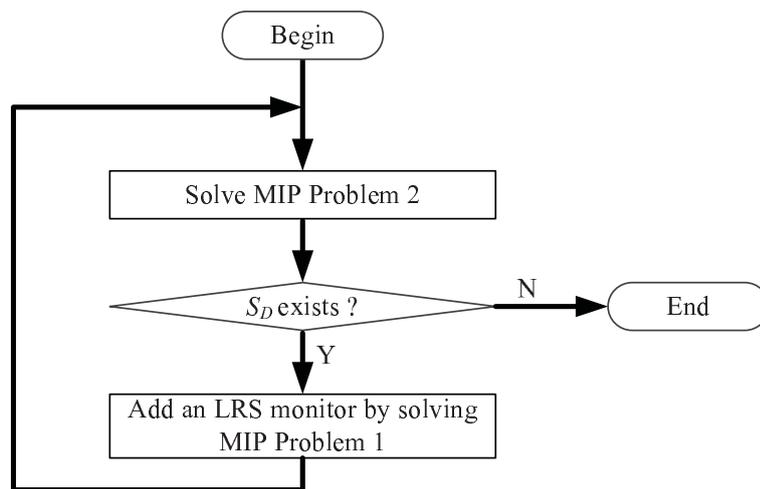


FIG. 7.2 – A schematic diagram of the iterative method.

A schematic diagram of the iterative method is shown in Fig. 7.2. First, Problem 2 is solved to find an MIMS S_D given a WS³PR. Next, a shared resource place r_i in the siphon with larger $M_0(r_i)$

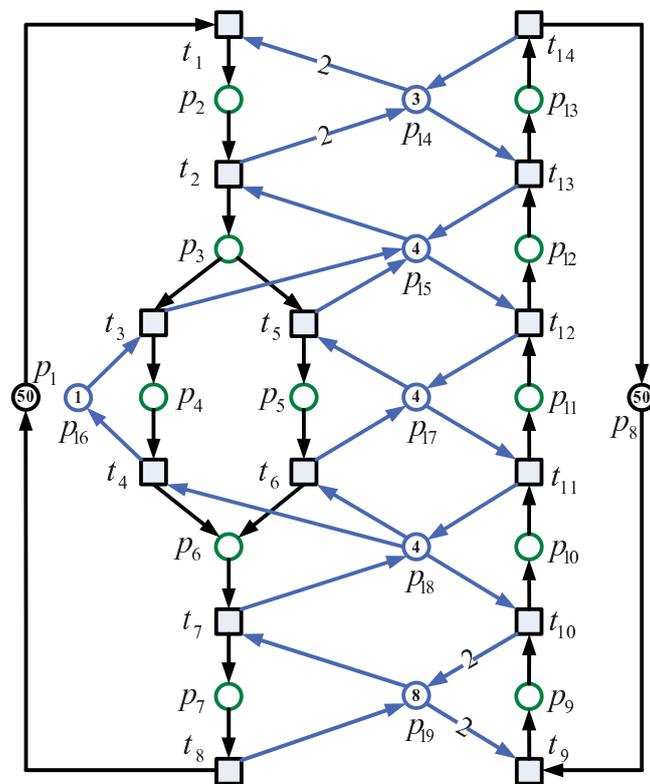


FIG. 7.3 – A WS³PR net model.

and less in and out-degree than others is chosen to be controlled by an LRS monitor v_i . A larger $M_0(r_i)$ implies that less reachable states will be lost after the addition of an LRS monitor, and less arcs mean that its calculation is easier. The two steps are iteratively executed until no MIMS can be found.

Being different from other methods using MIP, this proposed one does not induce new problematic siphons after adding LRS monitors since their connections are exactly the same with the resource places to be controlled. Hence, no induced-siphon needs extra control. Additionally, the MIMS found in an iteration may not be completely controlled by an LRS monitor designed for a resource place in the siphon. It only controls all WSDCs containing the resource place. All strict minimal siphons are minimally controlled when all iterations are finished. Since the number of LRS monitors for a WS^3PR is bounded by the number of its resource places, i.e., $|P_R|$, so is the number of iterations.

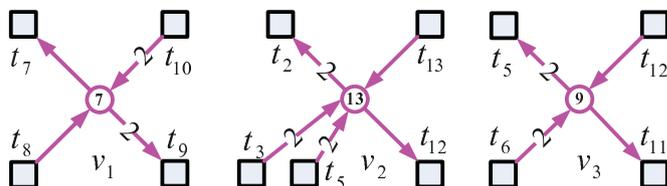


FIG. 7.4 – Three LRS monitors.

For example, a WS^3PR net model is shown in Fig. 7.3. After Problem 2 is solved, an MIMS can be obtained. A shared resource place p_{19} is chosen and a designed LRS monitor v_1 is imposed to control all minimal insufficiently marked siphons containing p_{19} . After three iterations, the controlled net model is live and with about 3/4 original reachable states reserved. Three LRS monitors designed in each iteration are shown in Fig. 7.4. More details can be found in Table. 7.1.

7.4. SUMMARY

TABLE 7.1 – Details of the iterative control of the WS³PR net shown in Fig. 7.3.

MIMS obtained by MIP	r_i	v_i	reachable states	dead states
/	/	/	3,334,653	30
$\{p_5, p_7, p_{10}, p_{12}, p_{13}, p_{15}, p_{16}, p_{17}, p_{18}, p_{19}\}$	p_{19}	v_1	2,663,888	6
$\{p_6, p_7, p_{11}, p_{12}, p_{13}, p_{15}, p_{16}, p_{17}, p_{18}\}$	p_{15}	v_2	2,613,824	1
$\{p_6, p_7, p_{11}, p_{12}, p_{13}, p_{16}, p_{17}, p_{18}\}$	p_{17}	v_3	2,500,037	0

7.4 Summary

The chapter presents an iterative liveness-enforcing method by adopting both MIP and LRS. Without the enumeration of all WSDC, it can achieve liveness within a fixed and known number of iterations, and all strict minimal siphons in the controlled net are minimally controlled. The proposed method can be extended and used in more complex plant models after a further investigation of circular waits of resources and the corresponding design of LRS monitors. The following problems are open : (1) How to optimally select a shared resource place given a maximal insufficiently marked siphon in an iteration, which leads to the minimal number of LRS monitors and the lowest computational cost, and (2) How to extend this method to more general nets than WS³PR, which demands a modified design of LRS monitors.

7.4. SUMMARY

Chapitre 8

On Intrinsically Live Structure of GS³PR Modeling FMS

The deadlock control and liveness-enforcement are mostly implemented based on siphons in most existing work using structural theory. Chapter 3 has revealed a kind of intrinsically live structures (ILS) in a class of generalized Petri nets called systems of simple sequential processes with weighted resources allocation (WS³PR). The liveness-enforcement is achieved by analyzing the structure including weight information and reconfiguring the initial marking of resource places without considering the initial token counts in idle process places. This chapter extends it to a more general class than WS³PR. Instead of siphons, a new concept of structural objects is defined to carry weight information and reflect the circular waits of resources. The numerical relationship between initial markings of resource places and weights of arcs is investigated to yield an ILS-based liveness-enforcing method that requires no external monitors. The work in this chapter provides an insight into structures of generalized Petri nets and a new revenue to achieve the desired deadlock control. Several examples are used to illustrate the proposed method.

8.1 Introduction

Inspired by the concept of circular waits and circular blockings and their relations with deadlock-freedom [52], Chapter 3 proposes ILSs of WS³PR. A new structural object, weighted simple direc-

ted circuit (WSDC), is presented to overcome the shortcoming of siphons that do not carry weight information. The liveness of a WS³PR can be enforced by checking its structures and reconfiguring its initial marking of resource places to make every WSDC satisfy a proposed restriction. In Chapter 4, ILSs are adopted to yield a hybrid method to identify and remove redundant monitors designed for elementary siphons - a concept pioneered in [53]. In Chapter 5, they are used to design a new kind of supervisors that enforce liveness through enforcing resource usage ratios of partial operations.

In this chapter, the ILS-based method is extended to a more general class of generalized Petri nets. Different from the results in previous chapters, some activity places may be contained in the weighted digraph that represents the wait relations of resources places. This is because of the more complex resource usage style of the class under consideration. These special activity places play a role as temporal resources due to their requiring, holding, and releasing the real system resources. Subsequently, the concept of extended WSDCs and competition paths are provided to characterize the intrinsically live structures of net models. A sufficient condition for liveness-enforcement is given, i.e., the liveness can be ensured by reconfiguring the initial marking of some resource places if the intrinsically live structures exist in a net model. The work in the chapter can improve the behavioral permissiveness and save the control cost, and furthermore, it provides a deeper insight into structures of generalized Petri nets.

8.2 GS³PR

The class of Petri nets under consideration in the chapter, general S³PR (GS³PR), is an extension to S³PR. WS³PR and System of Simple Sequential Processes with Multiple Resources (S³PMR) [37] are both subclasses of GS³PR that is a subclass of extended S³PR (ES³PR) [92]. The relation among them is shown in Fig. 8.1.

If $t_1, p_1, \dots, t_k, p_k, t_{k+1} \in P_A \cup T$ is a simple path, p_1, p_2, \dots, p_k are called consecutive activity places. For example, in Fig. 8.2(a), t_1, p_2, t_2, p_3, t_3 is a simple path, p_2 and p_3 are consecutive activity places. If there is only one place in the simple path, we also say it is consecutive by itself.

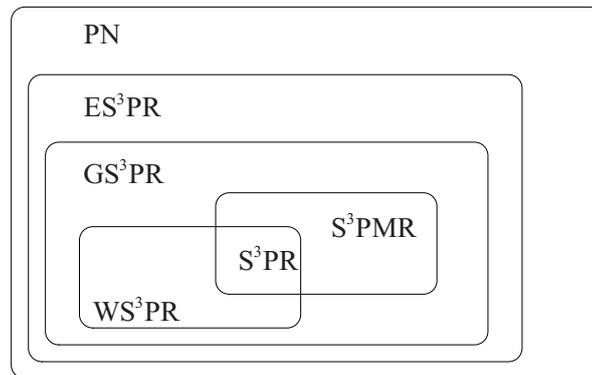


FIG. 8.1 – Classes of Petri nets.

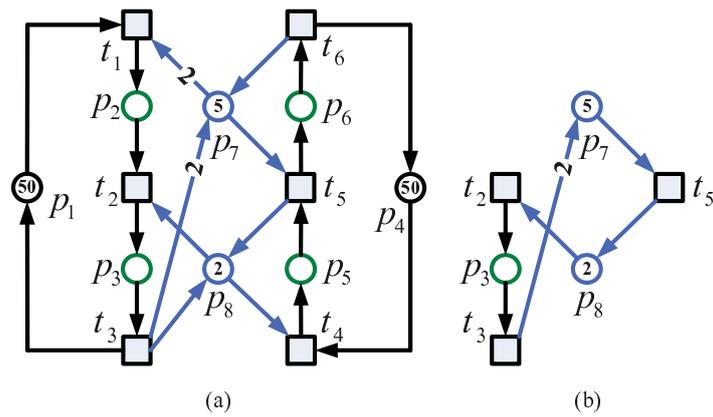


FIG. 8.2 – (a) A Petri net model and (b) one of its subnets.

Definition 8.1 A general simple sequential process with resources (GS²PR) is a generalized Petri net $N = (P_A \cup P_0 \cup P_R, T, F, W)$ such that

- (1) The subnet generated by $X = P_A \cup P_0 \cup T$ is an S²P, where $P_0 = \{p_0\}$;
- (2) $P_R \neq \emptyset$ is called a set of resource places with $(P_A \cup P_0) \cap P_R = \emptyset$, and each resource place $r \in P_R$ is associated with a nonempty set of consecutive activity places, $H(r)$, which implies that these operations require r , and the activity places in $H(r)$ are called holders of r and $P_A = \cup_{r \in P_R} H(r)$;
- (3) For each input transition t of a holder $p \in H(r)$, there exists an arc (r, t) from r to t if $\bullet t \cap H(r) = \emptyset$;
- (4) For each output transition t' of a holder $p \in H(r)$, there exists an arc (t', r) from t' to r if $t' \bullet \cap H(r) = \emptyset$;
- (5) $W = W_A \cup W_R$, where $W_A = ((P_A \cup P_0) \times T) \cup (T \times (P_A \cup P_0)) \rightarrow \{0, 1\}$ and $W_R = (P_R \times T) \cup (T \times P_R) \rightarrow \mathbb{N}$; and
- (6) $\bullet\bullet(P_0) \cap P_R = (P_0) \bullet\bullet \cap P_R = \emptyset$. If $t \in P_0^\bullet$ (resp. $t \in \bullet P_0$), t is called a source (resp. sink) transition.

Definition 8.2 Let $t_1, p_1, \dots, t_i, p_i, \dots, t_k, p_k, t_{k+1}$ be a path of N with an appropriate labeling and consecutive activity places $p_1, \dots, p_i, \dots, p_k$ be in a set of resource place r 's holders $H(r)$. It is said that (r, t_1) and (t_{k+1}, r) form an arc pair if they exist and there is no arc between t_i and r .

As shown in Fig. 8.3(a), (r, t_1) and (t_{k+1}, r) form an arc pair, and the group of consecutive holders $p_1, \dots, p_i, \dots, p_k$ require resources only by the first one and release them only by the last one. According to Definition 8.1, any other activity place between the first one and the last one of the consecutive ones cannot require and release resources related to r . Two examples of resource usage styles not permitted in GS²PR are given in Fig. 8.3(b). Note that (1) two arcs in an arc pair must have the same weights to guarantee the conservativeness and (2) both in Fig. 8.3(b) can be converted or arcs rearranged to become GS²PR.

Definition 8.3 Let $N = (P_A \cup P_0 \cup P_R, T, F, W)$ be a GS²PR. An initial marking M_0 is said to be acceptable if

- (1) $\forall p_0 \in P_0, M_0(p_0) \geq 1$;
- (2) $\forall p \in P_A, M_0(p) = 0$; and
- (3) $\forall r \in P_R, \forall t \in r^\bullet, M_0(r) \geq W(r, t)$.

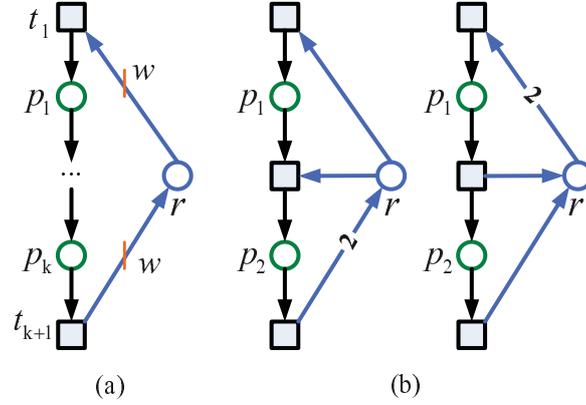


FIG. 8.3 – (a) An arc pair formed by (r, t_1) and (t_{k+1}, r) with $k \geq 1$ and (b) two examples of resource usage styles not permitted in GS²PR.

Definition 8.4 A system of GS²PR, called GS³PR for short, is defined recursively as follows

(1) A GS²PR is a GS³PR; and

(2) Let $N_i = (P_{A_i} \cup P_{0_i} \cup P_{R_i}, T_i, F_i, W_i)$, $i \in \{1, 2\}$, be two GS³PR such that $(P_{A_1} \cup P_{0_1}) \cap (P_{A_2} \cup P_{0_2}) = \emptyset$, $P_{R_1} \cap P_{R_2} = P_C \neq \emptyset$, and $T_1 \cap T_2 = \emptyset$. Then, the net $N = (P_A \cup P_0 \cup P_R, T, F, W)$ resulting from the composition of N_1 and N_2 via P_C (denoted by $N_1 \circ N_2$) such that (a) $P = P_{A_1} \cup P_{A_2}$, (b) $P_0 = P_{0_1} \cup P_{0_2}$, (c) $P_R = P_{R_1} \cup P_{R_2}$, (d) $T = T_1 \cup T_2$, (e) $F = F_1 \cup F_2$, and (f) $W = W_1 \cup W_2$ is also a GS³PR.

FMS modeled with Petri nets can be classified into three categories according to the number and type of resources required when a workpiece is to be processed at an operation step : (1) only one unit of a single type of resources, (2) one or multiple units of a single type of resources, and (3) one or multiple units of one or multiple types of resources. S³PR [16], WS³PR [64], and ES³PR [92] are adopted to model and analyze the three situations, respectively. Note that GS³PR is a subclass of ES³PR that allows the resource usage style shown in Fig. 8.3(b).

For example, the net model as shown in Fig. 8.2(a) is a GS³PR net model with idle process places $\{p_1, p_4\}$, activity places $\{p_2, p_3, p_5, p_6\}$, resource places $\{p_7, p_8\}$, and an acceptable initial marking $M_0 = 50p_1 + 50p_4 + 5p_7 + 2p_8$. The sets of p_7 's and p_8 's holders are $H(p_7) = \{p_2, p_3, p_6\}$ and $H(p_8) = \{p_3, p_5\}$, respectively. (p_7, t_1) and (t_3, p_7) , (p_7, t_5) and (t_6, p_7) , (p_8, t_2) and (t_3, p_8) , and (p_8, t_4) and (t_5, p_8) are four arc pairs. It is live and with 50 reachable states. Note that no matter how

many tokens (any acceptable initial markings) are initially configured in the idle process places, the net is always live. It is decided by a kind of intrinsically live structures and a special numerical relationship between arc weights and initial marking of resource places.

8.3 ILS in GS³PR

In Chapter 3, the ILS is proposed in WS³PR. The liveness of WS³PR can be enforced by analyzing the numerical relation among arc weights and reconfiguring the initial marking of resource places if ILS exists in the model. In Chapter 4, ILSs and elementary siphons [53] are combined together to yield a hybrid liveness-enforcing method to identify and remove redundant monitors for WS³PR. Furthermore, ILS is fully investigated in Chapter 5 and used to design a new kind of supervisors that achieve the goal of liveness-enforcement by enforcing resource usage ratios of some activity places. In this chapter, they are extended within the scope of GS³PR, which starts with the concept of weighted digraphs of wait relations.

Definition 8.5 Given GS³PR $N = (P_A \cup P_0 \cup P_R, T, F, W)$, a weighted digraph of wait relations, denoted as $G_{WR} = (P'_A \cup P'_R, T_R, F', W')$, is a subnet of N such that : (1) $P'_A \subseteq P_A$ and $P'_R \subseteq P_R$; (2) $T_R = \bullet(P'_A \cup P'_R) \cap (P'_A \cup P'_R)^\bullet$; (3) $F' = [((P'_A \cup P'_R) \times T_R) \cup (T_R \times (P'_A \cup P'_R))] \cap F$; and (4) $\forall x, y \in P'_A \cup P'_R \cup T_R, x^\bullet \neq \emptyset, \bullet x \neq \emptyset$, and $W'(x, y) = W(x, y)$.

=====

Algorithm 8.1

Derivation of the weighted digraph of wait relations of a GS³PR.

Input : A GS³PR $N = (P_0 \cup P_A \cup P_R, T, F, W)$.

Output : The weighted digraph of wait relation $G_{WR} = (P'_A \cup P'_R, T_R, F', W')$.

Step 1 : Remove all idle process places and arcs connecting to them from N ;

Step 2 : $\forall p \in P_A$, remove p and arcs connecting to it from N if it uses only one type of resources r and there exists an arc $(r, t), t \in \bullet p \cup p^\bullet$;

Step 3 : $\forall t \in T$, remove t and arcs connecting to it from N if its preset or postset is empty ;

Step 4 : For every remaining place, remove it and arcs connecting to it from N if its preset or postset is empty ;

Step 5 : G_{WR} is the remaining part of N .

=====

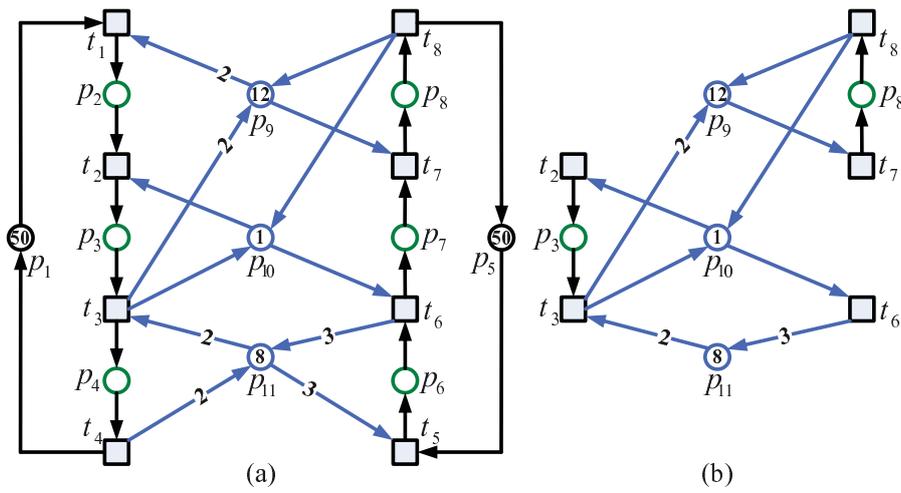


FIG. 8.4 – (a) A marked GS³PR net model and (b) its weighted digraph of wait relations.

In the previous chapters, all wait relations of resources in a WS³PR are represented by a weighted digraph that consists of only resource places. However, in GS³PR, the resource usage style becomes more complex than that of WS³PR. Some activity places using multiple types of resources act as a role of temporary resources that should be presented in the weighted digraph of wait relations¹. For example, GS³PR in Fig. 8.4(a) has idle process places $\{p_1, p_5\}$, activity places $\{p_2 - p_4, p_6 - p_8\}$, resource places $\{p_9 - p_{11}\}$ and an acceptable initial marking $M_0 = 50p_1 + 50p_5 + 12p_9 + p_{10} + 8p_{11}$. There are three strict minimal siphons, $S_1 = \{p_3, p_8, p_9, p_{10}\}$, $S_2 = \{p_4, p_7, p_8, p_{10}, p_{11}\}$, and $S_3 = \{p_4, p_8, p_9, p_{10}, p_{11}\}$. According to Algorithm 8.1, its weighted digraph of wait relations is derived as shown in Fig. 8.4(b). For example, activity place p_3 is kept in the subnet since it is involved in the wait relation between resource places p_9 and p_{10} . Next, a simple circuit is adopted to characterize ILS.

Definition 8.6 A weighted simple directed circuit (WSDC) in GS³PR, denoted as $c_W = (P^C, T_R^C, F^C, W^C)$,

¹It corresponds to and is compatible with the concept of weighted digraph of resource places in WS³PR.

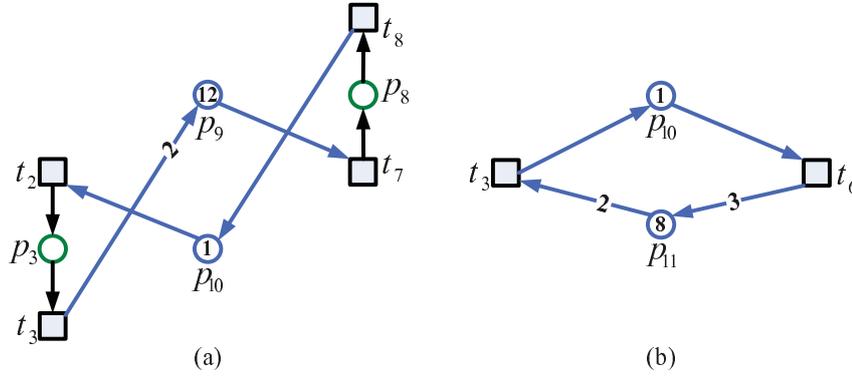


FIG. 8.5 – Two WSDCs of the GS³PR net shown in Fig. 8.4(a) : (a) a WSDC including activity places, and (b) another WSDC not including any activity place.

is a simple circuit derived from a weighted digraph of wait relations $G_{WR} = (P'_A \cup P'_R, T_R, F', W')$ such that :

- (1) $P^C = \{p_1, p_2, \dots, p_m\} \subseteq P'_A \cup P'_R$ ($m \geq 2$) and $|P^C \cap P'_R| \geq 2$;
- (2) $T_R^C = \{t_1, t_2, \dots, t_m\} \subseteq T_R$ ($m \geq 2$), $p_1^\bullet \cap \bullet p_2 = \{t_2\}$, and $p_2^\bullet \cap \bullet p_3 = \{t_3\}, \dots, p_m^\bullet \cap \bullet p_1 = \{t_1\}$;
- (3) $F^C = [(P^C \times T_R^C) \cup (T_R^C \times P^C)] \cap F'$; and
- (4) $W^C \subseteq W'$ and $W^C = W(x, y), \forall x, y \in P^C \cup T_R^C$.

WSDCs are a kind of structural objects explicitly carrying weight information and intuitively reflecting circular wait of resources. WSDC models a chain of resource places that form a circular wait among each other. Different from that in WS³PR, WSDC defined in GS³PR also contains some activity places that build an indirect circular wait relation among resource places. Note that not all the simple circuits found in the weighted digraph of wait relations are WSDCs by using the algorithm in [43]. The simple circuits containing at most one resource place do not represent a real circular wait and an initially configurable structure since that there is no token in activity places at the initial marking. For example, four simple circuits can be found from the weighted digraph of wait relations shown in Fig. 8.4(b), $c_1 = t_2 p_3 t_3 p_9 t_7 p_8 t_8 p_{10} t_2$, $c_2 = t_3 p_{10} t_6 p_{11} t_3$, $c_3 = t_8 p_9 t_7 p_8 t_8$, and $c_4 = t_2 p_3 t_3 p_{10} t_2$. According to Definition 8.6, c_1 and c_2 are two WSDCs as shown in Fig. 8.5. Note that activity places p_3 and p_8 are included in c_1 and make connections of the circular wait relation between resource places p_9 and p_{10} . A release of p_9 follows a requirement of p_{10} and

a release by p_3 . Both c_3 and c_4 contain one resource place and one activity one and are not real WSDCs.

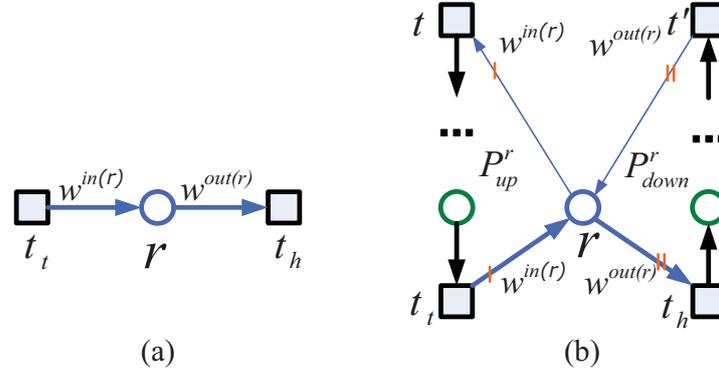


FIG. 8.6 – (a) A competition path $t_t r t_h$ and (b) its sets of upstream and downstream activity places P_{up}^r and P_{down}^r .

Theorem 8.1

Given a WSDC $c_W = (P^C, T_R^C, F^C, W^C)$ of a GS³PR $N = (P_A \cup P_0 \cup P_R, T, F, W)$, $\bigcup_{r \in (P^C \cap P_R)} P_{down}^r \cup P^C$ is a strict minimal siphon, where $P_{down}^r \subseteq H(r)$ and is called the set of downstream activity places of r which are consecutive places such that there exists an arc (r, t) , $t \in \bullet P_{down}^r$.

For example, all places of c_2 in Fig. 8.5(b) and the downstream activity places $P_{down}^{p_{11}} = \{p_4\}$ and $P_{down}^{p_{10}} = \{p_7, p_8\}$ together form siphon S_2 . It is said that a WSDC is contained in a siphon if all places of the WSDC are included in the siphon. Note that a siphon may contain several WSDCs even if their places are the same. In the example, c_1 and c_2 are contained by siphons S_1 and S_2 , respectively.

Definition 8.7 A simple path $l = (P_R^L, T_R^L, F_R^L, W_R^L)$ in GS³PR is called a resource path derived from WSDC $c_W = (P^C, T_R^C, F^C, W^C)$ such that :

- (1) $P_R^L = \{r\} \subseteq (P^C \cap P_R^C)$;
- (2) $T_R^L = \{t_t, t_h\}$, where there exist a tail transition t_t with $\{t_t\} = \bullet r \cap T_R^C$, and a head one t_h with $\{t_h\} = r \bullet \cap T_R^C$;
- (3) $F^L = \{(t_t, r), (r, t_h)\} \subseteq F^C$; and
- (4) $W_R^L = \{W(t_t, r), W(r, t_h)\} \subseteq W^C$; where (t_t, r) and (r, t_h) are called in-arc and out-arc of a resource path, respectively. If the in-arc and out-arc of a resource path do not form an arc pair, this

resource path is called a competition path, denoted as l_{cp} .

For simplicity, a resource/competition path is denoted by $t_t r t_h$, and the weights of the in-arc and the out-arc are denoted by $w^{in(r)}$ and $w^{out(r)}$ as shown in Fig. 8.6. Besides siphons, WSDCs are characterized as a supplement of structural objects connecting with deadlock-freedom and liveness in GS³PR. A WSDC represents a quantitative chain of circular wait with arc weights. More specifically, a competition path is a link of a WSDC and defined to represent a competition for tokens in the resource place between two sets of holders. As shown in Fig. 8.6(b), a set of upstream (resp. downstream) activity places P_{up}^r (resp. P_{down}^r) of a competition path $t_t r t_h$ is the set of consecutive activity places using resource place r , and (t_t, r) and (r, t_h) (resp. (t, r) and (r, t_h)) form an arc pair.

We use \mathbf{L}_W to denote a set of competition paths with the same resource place and in-arc weight larger than out-arc weight, i.e., $\mathbf{L}_W = \{l_{cp} = t_t r t_h | W(t_t, r) > W(r, t_h)\}$. A weight matrix $[\mathbf{W}]$ of all n competition paths in \mathbf{L}_W with respect to a resource place r is a $2 \times n$ matrix :

$$[\mathbf{W}^{(r)}] = \begin{bmatrix} \mathbf{W}^{in(r)} \\ \mathbf{W}^{out(r)} \end{bmatrix} = \begin{bmatrix} w_1^{in(r)} & \dots & w_i^{in(r)} & \dots & w_n^{in(r)} \\ w_1^{out(r)} & \dots & w_i^{out(r)} & \dots & w_n^{out(r)} \end{bmatrix},$$

where $w_i^{in(r)}$ (resp. $w_i^{out(r)}$) is the in-arc (resp. out-arc) weight of the i th competition path with r in \mathbf{L}_W , $w_i^{in(r)} \in \mathbb{N}^+ \setminus \{1\}$, $w_i^{out(r)} \in \mathbb{N}^+$, $n \in \mathbb{N}^+$, and $i \in \mathbb{N}_n$.

Based on the above concepts, the same restriction proposed for WS³PR in Chapter 3 still holds to characterize the numerical relation between arc weights and initial markings of resource places in ILS of GS³PR. It has the same meaning when we say that resource place r in a competition path $l_{cp} = t_t r t_h$ satisfies Restriction 8.1 or a WSDC containing l_{cp} satisfies Restriction 8.1 as given next.

Restriction 8.1

Given a marked GS³PR (N, M_0) , a resource place r satisfies the following two conditions :

- (1) $(M_0(r) \bmod w_i^{in(r)}) \geq w_i^{out(r)}$, where $M_0(r)$ is the initial marking of r , and $w_i^{in(r)}$ (resp. $w_i^{out(r)}$) is the in-arc (resp. out-arc) weight of the i th competition path in \mathbf{L}_W ; and
- (2) There exists no n -dimensional row vector $\mathbf{A} = [a_1 \dots a_i \dots a_n]$ such that $0 \leq (M_0(r) - \mathbf{A}[\mathbf{W}^{in(r)}]^T) < \min\{w_i^{out(r)}\}$, where $a_i \in \mathbb{N}$.

The essential idea behind Restriction 8.1 is to prevent all tokens in resource place r from being all used by holders only in the set of upstream activity places. It is implemented by imposing a limitation on the numerical relation between initial number of tokens in r and weights of its in-arc and out-arc. Therefore, a competition path of a WSDC satisfying Restriction 8.1 implies that a link of a chain of circular waiting is broken.

Theorem 8.2

Given a marked GS³PR (N, M_0) , a siphon S is minimally controlled if all WSDCs contained in it satisfy Restriction 8.1.

For example, in Fig. 8.4(a), S_2 is minimally controlled since the only WSDC c_2 contained by it satisfies Restriction 8.1, while S_1 is not due to the initial marking of p_9 , $M_0(p_9) = 12$. If it is reconfigured, i.e., changing $M_0(p_9) = 12$ into $M_0(p_9) = 11$, S_1 will also be minimally controlled.

Theorem 8.3

A marked GS³PR (N, M_0) is live if every WSDC satisfies Restriction 8.1.

Theorem 8.3 builds a bridge between the liveness of GS³PR and ILS. It improves the behavioral permissiveness of the controlled systems. In Table 8.1, a detailed comparison between control effects of elementary siphon-based methods (ES) in [28] [120] and ILS-based one in this work for GS³PR in Fig. 8.4(a) is provided. The original net has total 234 reachable states and one deadlock among them. S_1 and S_2 from all three strict minimal siphons are selected as elementary ones. Two monitors and corresponding arcs are designed to control them. The controlled net is live and has 124 reachable states. It can be found that ES is very conservative and loses almost a half number of states since the output arcs of monitors are all connected to source transitions to avoid introducing new problematic siphons. By adjusting the output arcs, it yields a live net with 168 reachable states, which is about 72.1% (168/233) of the GS³PR net's maximal permissiveness.

TABLE 8.1 – A Comparison between the control effects of the elementary siphons-based method (ES) in [120] and the intrinsically live structures-based one (ILS) in this chapter for the GS³PR net model shown in Fig. 8.4(a).

Description	$M_0(P_R)$	$M_0(V)$	Arcs and associated weights of monitors	$ R(N, M_0) $	Live
Original net : without control	$12p_9 + p_{10} + 8p_{11}$	/	/	234	No
ES by connecting output arcs of monitors to source transitions	$12p_9 + p_{10} + 8p_{11}$	$11v_1 + 5v_2$	$W(v_1, t_1) = W(t_2, v_1) = 2,$ $W(v_1, t_5) = W(t_7, v_1) = 1,$ $W(v_2, t_1) = W(t_3, v_2) = 1,$ $W(v_2, t_5) = W(t_6, v_2) = 3$	124	Yes
			and		
ES with a rearrangement of output arcs of monitors	$12p_9 + p_{10} + 8p_{11}$	$11v_1 + 5v_2$	$W(v_1, t_1) = W(t_2, v_1) = 2,$ $W(v_1, t_6) = W(t_7, v_1) = 1,$ $W(v_2, t_2) = W(t_3, v_2) = 1,$ $W(v_2, t_5) = W(t_6, v_2) = 3$	168	Yes
			and		
ILS by removing 1 token from p_9	$11p_9 + p_{10} + 8p_{11}$	/	/	206	Yes
ILS by adding 1 token in p_9	$13p_9 + p_{10} + 8p_{11}$	/	/	242	Yes

By applying the proposed method, the liveness-enforcement can be achieved by only reconfiguring the initial number of tokens in resource place p_9 . After removing one token from p_9 , the resulting new initial marking of all resource places $M_0(P_R) = 11p_9 + p_{10} + 8p_{11}$ makes two WSDCs both satisfy Restriction 8.1. Hence, the net is live and can reach 87.6% (206/233) of the maximal permissiveness. It implies that the control cost can be saved without imposing external monitors and corresponding arcs. Furthermore, it is not always effective to enforce liveness by increasing the quantity of system resources, contrary to the conventional wisdom. In this example, a new initial marking $M'_0(P_R) = 13p_9 + p_{10} + 8p_{11}$ can ensure the liveness, whereas $M''_0(P_R) = 14p_9 + p_{10} + 8p_{11}$ cannot. It is the special numerical relationship with proper resources that guarantees the liveness. Note that monitors that have the same control effect can be designed instead of decreasing the initial numbers of tokens in resource places. For example, a monitor v_1 with $M_0(v_1) = 11$, $W(v_1, t_1) = W(t_3, v_1) = 2$, and $W(v_1, t_7) = W(t_8, v_1) = 1$ can be added to the original net with $M_0(P_R) = 12p_9 + p_{10} + 8p_{11}$. It restrains one token in p_9 .

8.4 Summary

The work in the chapter extends the intrinsically live structures-based liveness-enforcing method that was first proposed in WS^3PR to a more general subclass of Petri nets, GS^3PR . A sufficient condition of liveness is given, i.e., the liveness-enforcement can be achieved by analyzing the arc weights and reconfiguring the initial marking of some resource places if such intrinsically live structures exist in GS^3PR . This method saves the control cost and improves the system behavioral permissiveness. Future work should address two challenging issues : (1) How to effectively find all real WSDCs from the weighted digraph of wait relations of GS^3PR and optimally select competition paths from them, which leads to the minimal reconfiguration of system resources and the optimal (or sub-optimal) behavioral permissiveness, and (2) How to design liveness and resource usage ratio-enforcing supervisors for GS^3PR based on the mechanism of intrinsically live structures, which depends on the further analysis of the relation between upstream and downstream activity places.

8.4. SUMMARY

Chapitre 9

ILS in Process and Resource-Oriented Petri Nets

As an effective and graphical mathematical tool, Petri nets are extensively used in the modeling, analysis, verification, and control of automated manufacturing systems. Process and resource-oriented Petri nets (POPNS and ROPNs) are successively proposed in literature focusing on the description of product process and resource usage sequence, respectively. Structural theory is applied on both to solve the deadlock control problem. In this work, a class of colored ROPNs (CROPNs) is defined corresponding to WS^3PR that is a class of POPNs. A kind of intrinsically live structures (ILSs) is investigated in both POPNs and CROPNs.

9.1 SRS – a class of ROPN

In this chapter, a class of ROPNs is presented according to WS^3PR in POPNs. Colors are introduced to distinguish different types of workpieces and control different production routes. In a comparative way with respect to that in WS^3PR in the previous chapters, ILSs in such ROPNs are explored and a numerical restriction is given to ensure the liveness of such ROPNs. This part can be considered as a first step to apply the ILS-based method to ROPNs.

A class of resource-oriented Petri nets, namely a System of Resource-oriented Route (SRS) is presented in the section to model a workpiece production process, which is viewed as a workpiece

visiting needed resources one by one according to a specification. Colors are introduced to distinguish different types of workpieces and control different production routes. It uses a place to model the unique input/output station, and a place for every resource. A transition models a process that a workpiece is transported between resources.

Definition 9.1 A resource-oriented sequence (RS) is a generalized finite colored Petri net $N = (\{p_0\} \cup P_R, T, F, W, \{c\})$, where (1) p_0 is called an idle process place with $K(p_0) = \infty$, (2) P_R is a set of resource places, and $\forall r \in P_R, K(r) \in \mathbb{N}^+$, (3) N is a simple circuit of all nodes in $\{p_0\} \cup P_R \cup T$, (4) $\forall r \in P_R, t \in \bullet r, t' \in r \bullet W(t, r) = W(r, t')$, and each transition is associated with a unique color c .

$K(p_0) = \infty$ means that p_0 is a loading/unloading station and has no capacity constraint. We use $M(p)(c)$ to denote the number of tokens with color c in place p , whereas $M(p)$ to denote that of all tokens in p . Let $N = (\{p_0\} \cup P_R, T, F, W, \{c\})$ be an RS. An initial marking M_0 is said to be acceptable if (1) all tokens in $p_0 \in P_0$ are associated with color c , (2) $M_0(p_0) = M_0(p_0)(c) > 0$, and (3) $\forall r \in P_R, M_0(r) = 0$.

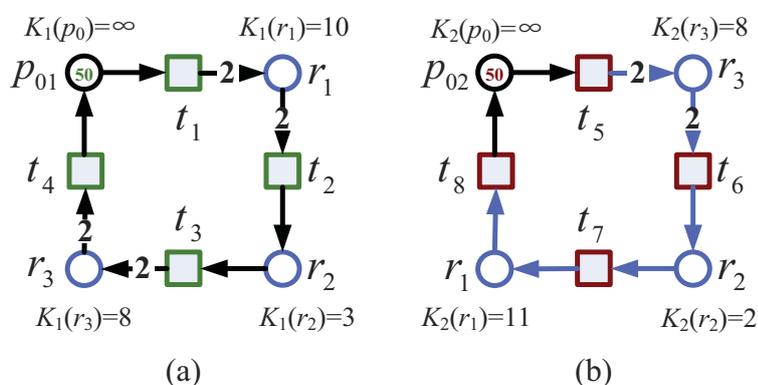


FIG. 9.1 – (a) An RS N_1 processing A-type of workpieces and (b) another RS N_2 processing B-type of workpieces.

Consider the example in Fig. 9.1, types A and B workpieces are processed in two RSs, which are described as $p_{01} \rightarrow t_1 \rightarrow r_1 \rightarrow t_2 \rightarrow r_2 \rightarrow t_3 \rightarrow r_3 \rightarrow p_{01}$ and $p_{02} \rightarrow t_5 \rightarrow r_3 \rightarrow t_6 \rightarrow r_2 \rightarrow t_7 \rightarrow r_1 \rightarrow t_8 \rightarrow p_{02}$, respectively. It means that the sequence of resource requirement of A-type (resp. B-type) of workpieces is r_1, r_2 , and r_3 (resp. r_3, r_2 , and r_1). All transitions in N_1 (or N_2) are associated with a unique color, and the tokens initially allocated in the idle process

place are with the same color. The property of color can be implemented by radio-frequency identification (RFID) or smart card in a real system [114]. It is used to distinguish and control the process routes of different types of workpieces in a system of RSs, which is explained later. In p_{01} (or p_{02}), we initially put 50 tokens meaning that we do not limit $M_0(p_{01})$ (or $M_0(p_{02})$). In N_1 , $W(t_1, r_1)$ represents that a workpiece needs two slots of r_1 and $W(t_1, r_1) = W(r_1, t_2) = 2$ ensures the conservativeness. Note that even for the same resource, different capacities are used in different RSs. In N_2 , $K_1(r_1) = 11$, whereas $K_2(r_1) = 10$ in N_1 since the weights of its in-arc and out-arc equal to two and $K_2(r_1) = 11$ does not make sense. In other words, the capacity of a resource can be increased or decreased according to different specifications.

Definition 9.2 A system of resource-oriented sequences (SRS) is defined recursively as follows :
 (1) An RS is an SRS ; (2) Let $N_i = (\{p_{0i}\} \cup P_{Ri}, T_i, F_i, W_i, \{c_i\})$, $i \in \{1, 2\}$ be two RSs such that $P_{R1} \cap P_{R2} \neq \emptyset$, $T_1 \cap T_2 = \emptyset$, and c_1 is different from c_2 . Then, the net $N = (\{p_0\} \cup P_R, T, F, W, \mathbb{C})$ from merging N_1 and N_2 via their common resource places such that (a) p_0 is the unique idle process places with $K(p_0) = \infty$, (b) $P_R = P_{R1} \cup P_{R2}$, $\forall r \in P_{R1} \cup P_{R2}$, $K(r) = \max\{K_1(r), K_2(r)\}$, (c) $T = T_1 \cup T_2$ and every transition remains its original color, (d) $F = F_1 \cup F_2$, (e) $W = W_1 \cup W_2$, and (f) $\mathbb{C} = \{c_1\} \cup \{c_2\}$ is also an SRS.

Let $M_{01} = M_{01}(p_{01})$ and $M_{02} = M_{01}(p_{02})$ be the initial markings of N_1 and N_2 , respectively. The initial marking of N is $M_0 = M_0(p_0)p_0$ with $M_0(p_0) = M_0(p_0)(c_1) + M_0(p_0)(c_2) = M_{01}(p_{01}) + M_{02}(p_{02})$.

As shown in Fig. 9.2, an SRS is obtained by merging N_1 and N_2 in Figs. 9.1(a) and (b), respectively. In p_0 , two types of tokens with color c_1 and c_2 are both contained. The tokens can only enable transitions with the same color. For example, if there are totally three tokens in r_1 including one in color c_1 and the other two in color c_2 , t_8 is enabled but t_2 is not since $M(r_1)(c_1) < 2$. Different colored tokens compete against each other for the slots of resource places. Note that during the merging, the relatively large capacity of a resource place in an RS is kept in the SRS. An intrinsically live structure with a special numerical relationship between arc weights and the capacities of resource places is discussed in the following section.

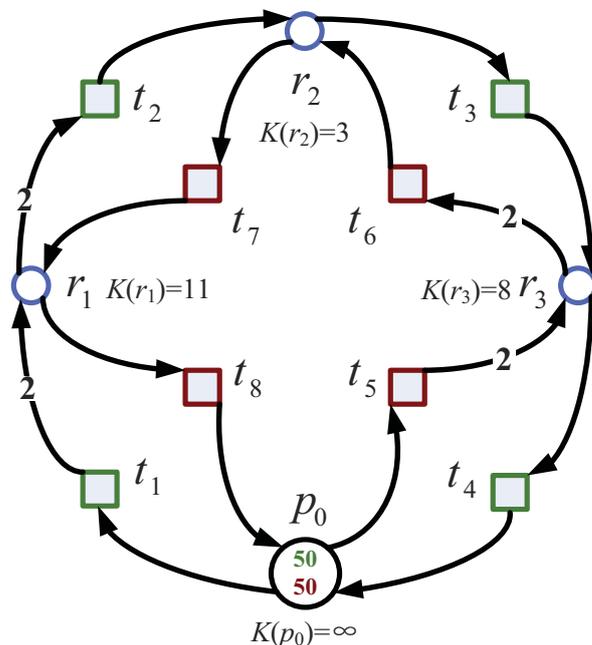


FIG. 9.2 – An SRS obtained by merging N_1 and N_2 in Fig. 9.1.

9.2 ILS in SRS

In order to maintain the consistency, we use the same concepts about ILS in SRS as we did in WS³PR. The definitions of weighted digraphs of resource places, WSDCs, and competition paths are given as follows.

Definition 9.3 Given an SRS $N = (\{p_0\} \cup P_R, T, F, W, \mathbb{C})$, a weighted digraph of resource places, denoted as $G_R = (P'_R, T_R, F_R, W_R, \mathbb{C})$, is a subnet of N such that : (1) $P'_R \subseteq P_R$; (2) $T_R = \bullet P'_R \cap P'_R \bullet$; (3) $F_R = [(P'_R \times T_R) \cup (T_R \times P'_R)] \cap F$; and (4) $W_R = W(x, y), \forall x, y \in P'_R \cup T_R$.

An SRS is a state machine, and a weighted digraph of resource places derived from the SRS is the maximal strongly connected component of resource places and transitions. For example, the weighted digraph of resource places of the SRS in Fig. 9.2 is shown in Fig. 9.3(a). It can be obtained by first removing the idle process place and connecting arc and then all resulting nodes only with in-arc or out-arc until none of them exists. It reflects the competition relation among all shared resource places.

Definition 9.4 A WSDC in an SRS, denoted as $c_W = (P_R^C, T_R^C, F_R^C, W_R^C, \mathbb{C}^C)$, is a simple circuit derived from a weighted digraph of resource places $G_R = (P_R', T_R, F_R, W_R, \mathbb{C})$ such that : (1) $P_R^C = \{r_1, r_2, \dots, r_m\} \subseteq P_R'$, $T_R^C = \{t_1, t_2, \dots, t_m\} \subseteq T_R$ ($m \geq 2$), $r_1^\bullet \cap \bullet r_2 = \{t_2\}$, and $r_2^\bullet \cap \bullet r_3 = \{t_3\}$, $\dots, r_m^\bullet \cap \bullet r_1 = \{t_1\}$; (2) $F_R^C = [(P_R^C \times T_R^C) \cup (T_R^C \times P_R^C)] \cap F_R$; (3) $W_R^C = W(x, y), \forall x, y \in P_R^C \cup T_R^C$; and (4) $\mathbb{C}^C \subseteq \mathbb{C}$.

Definition 9.5 A simple path $l = (P_R^L, T_R^L, F_R^L, W_R^L, \mathbb{C}^L)$ is called a resource path derived from a WSDC $c_W = (P_R^C, T_R^C, F_R^C, W_R^C, \mathbb{C}^C)$ such that : (1) $P_R^L = \{r\} \subseteq P_R^C$; (2) $T_R^L = \{t_t, t_h\}$, where there exist a tail transition t_t with $\{t_t\} = \bullet r \cap T_R^C$, and a head one t_h with $\{t_h\} = r^\bullet \cap T_R^C$; (3) $F_R^L = \{(t_t, r), (r, t_h)\} \subseteq F_R^C$; (4) $W_R^L = \{W(t_t, r), W(r, t_h)\} \subseteq W_R^C$; where (t_t, r) and (r, t_h) are called its in-arc and out-arc, respectively; and (5) $\mathbb{C}^L \subseteq \mathbb{C}^C$ and $\mathbb{C} = \{c_{t_t}, c_{t_h}\}$. If c_{t_t} is different from c_{t_h} , this resource path is called a competition path, denoted as l_{cp} .

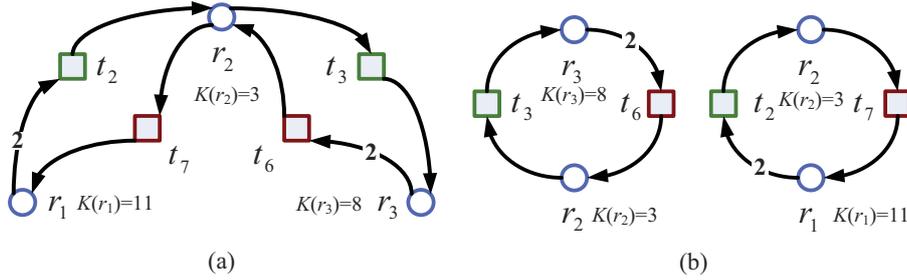


FIG. 9.3 – (a) A weighted digraph of resource places of the SRS in Fig. 9.2 and (b) two WSDCs derived from the SRS.

For example, two WSDCs as shown in Fig. 9.3(b) can be derived from the weighted digraph of resource places in Fig. 9.3(a). There are four competition paths in total, $t_3r_3t_6$, $t_6r_2t_3$, $t_2r_2t_7$, and $t_7r_1t_2$. Note that the in-arc and out-arc of a competition path actually model two types of tokens entering the resource place in a compact way. A competition path reflects a competition for the limited capacity of a resource place. Similar to the situation in WS^3PR , a weight matrix $[W^{(r)}] = [W^{in(r)} \ W^{out(r)}]^T$ can be defined to express the weight information of n competition paths with the same resource place r .

Restriction 9.1

Given a marked SRS (N, M_0) , a resource place r satisfies (1) $(K(r) \bmod w_i^{out(r)}) \geq w_i^{in(r)}$, where $K(r)$ is the capacity of resource place r , $w_i^{in(r)}$ (resp. $w_i^{out(r)}$) is the in-arc (resp. out-arc) weight

of one of all n competition paths with r ; and (2) there exists no n -dimensional row vector $\mathbf{B} = [b_1 \dots b_i \dots b_n]$ such that $0 \leq (K(r) - \mathbf{B}[\mathbf{W}^{out(r)}]^T) < \min\{w_i^{in(r)}\}$, where $b_i \in \mathbb{N}$.

With the same mechanism, Restriction 9.1 is just another expression form of Restriction 3.1. Being more intuitive, it means that the remaining capacity of a resource place after a prioritized and maximal entering of a type of tokens represented by the out-arc is still big enough to accommodate once entering of the other type of tokens represented by the in-arc. If a WSDC contains a resource place satisfying Restriction 9.1, it is said that the WSDC satisfies Restriction 9.1. A potential chain of circular waits is prevented by applying Restriction 9.1 to break one link of the chain. The following result can be given if all WSDCs are considered with Restriction 9.1.

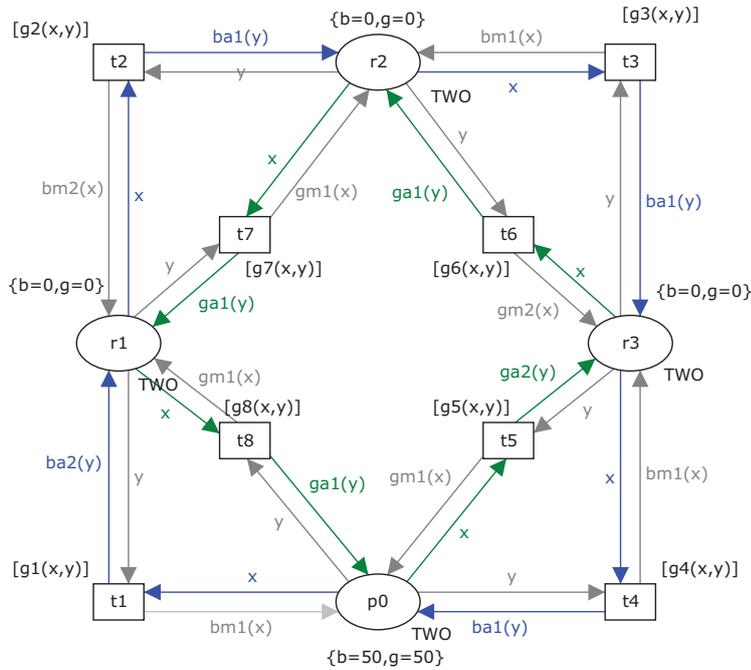


FIG. 9.4 – A CPN-Tools [7] model of the SRS in Fig. 9.2.

Theorem 9.1

A marked SRS (N, M_0) is live if every WSDC satisfies Restriction 9.1.

Theorem 9.1 makes the ILS-based liveness-enforcing method feasible in SRS. When applying this method, we check the structure of an SRS net model first, and then adjust the capacities of some certain resource places. A deadlock prevention can be achieved without any external

supervisor.

For example, we consider the SRS net model shown in Fig. 9.2. It has 10,353 reachable states in total and is not live. Two WSDCs, $t_3r_3t_6r_2t_3$ and $t_2r_2t_7r_1t_2$, are derived from its weighted digraph of resource places as shown in Figs. ??(b). It can be found that the latter satisfies Restriction 9.1, whereas the former does not. By freezing a part of capacity of r_3 , i.e., $K(r_3) = 7$, we can make the former satisfy Restriction 9.1. The net model is live with 8,263 reachable states. Alternatively, we may increase the capacity of r_3 , i.e., $K(r_3) = 9$. Then it satisfies Restriction 9.1 and the net model is live with 12,443 reachable states.

9.3 Summary

In this work, a class of colored resource-oriented Petri nets, SRS, is defined corresponding to WS³PR, which provides a compact style to model FMS from a view point of sequences of required resources. The concept of ILSs is introduced into SRS, and then a deadlock prevention method based on structural analysis is proposed, which needs no external supervisor. The further work includes : (1) an investigation of resource usage ratios in SRS ; and (2) optimal control of SRS.

9.3. SUMMARY

Chapitre 10

Conclusions and Future Research

This chapter concludes the thesis. The main contributions of the work are recalled. Some open and interesting problems are outlined in future work.

10.1 Contributions

The thesis works on the intrinsically live structure (ILS) and deadlock control of a class of generalized Petri nets modeling FMSs. Different from the existing siphon-based methods, the concept of ILSs based on WSDCs and the proposed restrictions is a general notation and the kernel of a series of deadlock control and liveness-enforcing methods presented in the thesis. The characteristics and essence of ILSs are identified and derived from WS^3PR and GS^3PR to implement such methods. The numerical relationship between initial markings and weights of connecting arcs is thoroughly investigated and used to design restrictions that ensure the intrinsic liveness of global and local structures.

The research work in the thesis is shown in Fig. 10.1. The study of ILSs in Chapter 3 is the root of the whole work. Combining elementary siphons with the ILS-based method, a hybrid liveness-enforcing one is presented in Chapter 4. According to the essence of ILS, the resource usage ratios (RU-ratios) of activity places in WS^3PR are analyzed and employed to design a new kind of parameterized liveness and ratio-enforcing supervisors (LRSs) in Chapter 5. When the concept of ILSs is adopted to divide and simplify the structures of a WS^3PR and LRS is used to control each resulting subnet, this work produces a divide-and-conquer paradigm as presented

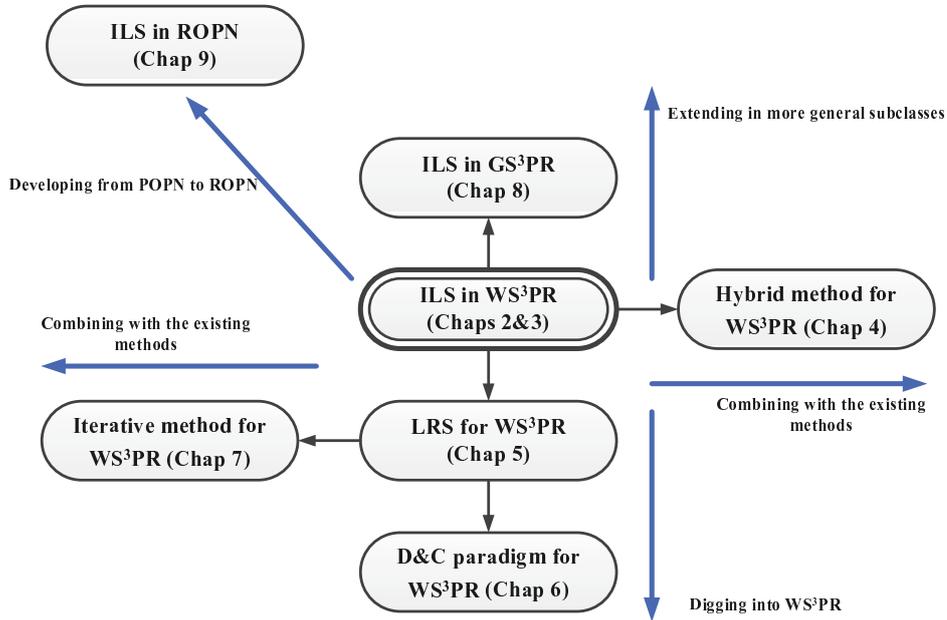


FIG. 10.1 – The main work of the thesis.

in Chapter 6. When LRS is combined with mixed integer programming (MIP), which is used to obtain a maximal insufficiently marked siphon (MIMS), this work proposes an iterative liveness-enforcing method as shown in Chapter 7. Chapter 8 explores the ILS in a more general class of Petri nets than WS^3PR , which is supposed to be an attempt to explore more complex structures. Finally, Chapter 9 extends the concept of ILSs from POPN to ROPN, which can be seen as a new beginning of the future work.

The main contributions of the work are summarized and highlighted chapter by chapter as follows.

(1) Chapter 3 : The ILS is proposed and fully investigated and an ILS-based liveness-enforcing method is presented. In some situations, the proposed method can make a system live by decreasing system resources instead of simply increasing them. A proper resource allocation is practically guaranteed by a proper numerical relationship between initial markings of resource places and arc weights. Compared with most existing deadlock prevention control policies, the new one does not add any monitors and arcs. The cost of its control software and hardware implementation can be saved. Meanwhile, the problem of structural complexity can be avoided.

(2) Chapter 4 : The ILS-based method is further developed. It is found that the identification of an ILS and calculation of proper initial number of tokens can be attributed to a typical congruence problem in number theory. A mathematical programming is designed to solve the problem in the chapter. Since the pure application of the ILS-based method demands that all WSDCs in a WS^3PR should first satisfy some structural conditions, which greatly limits its application scope, a hybrid liveness-enforcing method is proposed by combining the ILS-based one with the well-accepted elementary siphon-based one. The hybrid one no longer demands a global ILS of WS^3PR , and therefore, the application of WSDCs is extended. Theorems proved in this chapter build a bridge between WSDCs and siphons to realize this hybrid method. In addition, the minimally marked and controlled siphons are analyzed with the assistance of WSDCs. The hybrid method can simplify the liveness-enforcing supervisor for a system, improve the permissiveness, and finally, reduce the control implementation cost.

(3) Chapter 5 : The proposed LRS achieves liveness and ratio-enforcement at the same time. In particular, the liveness-enforcement is implemented by imposing a proper resource usage ratio-enforcement on a plant model. It is an attempt to pursue the precise usage of resources instead of maximal behavioral permissiveness. The number of monitors in such an LRS is bounded by that of resources used in the plant model. Hence, the control cost can be estimated without difficulty. In addition, the topology of the supervisor can be easily determined when a plant model is given, and is simple enough to be understood and adopted by industrial practitioners. When the initial configuration and weights of the plant model are changed, the proposed supervisor can adapt quickly by adjusting control parameters only. The connection of monitors remains unchanged. It resembles the work style of today's programable logic controllers, i.e., the re-hard-wired job can be avoided. This greatly saves control cost.

(4) Chapter 6 : The divide-and-conquer (D&C) liveness-enforcing method is investigated in WS^3PR . By fully utilizing the global and local ILSs, a WS^3PR is first considered under primary and advanced separations and then the remaining subnets are controlled by LRS. The way of breaking the control problem into sub-problems and the application of LRS are the kernel of the work in the chapter that provides a deeper insight into model structures and makes the analysis and control of complex WS^3PR feasible.

(5) Chapter 7 : MIP and LRS are combined together to yield an iterative liveness-enforcing method for WS^3PR . The application of LRS no longer depends on the enumeration of WSDCs but shared resource places in MIMSs obtained via MIP. The number of iterations can thus be limited. All strict minimal siphons in the controlled net model are minimally controlled. It lays a foundation for a direct application of LRS without the enumeration of WSDC in more general classes of Petri nets.

(6) Chapter 8 : The ILS-based method is extended to a more general class of generalized Petri nets. Different from the results in WS^3PR , some activity places may be contained in the weighted digraph that represents the wait relations of resources places. This is because of the more complex resource usage style of the class under consideration. These special activity places play a role as temporal resources due to their requiring, holding, and releasing the real system resources. Subsequently, the concept of extended WSDCs and competition paths are provided to characterize the ILSs of net models. A sufficient condition for liveness-enforcement is given, i.e., the liveness can be ensured by reconfiguring the initial marking of some resource places if the ILSs exist in a net model. The result can improve the behavioral permissiveness and save the control cost, and furthermore, it provides a deeper insight into structures of more general Petri nets.

(7) Chapter 9 : The concept of ILSs is extended from the framework of process-oriented Petri nets (POP_N) to that of resource-oriented Petri nets (ROP_N). ROP_N provides a compact style for modeling large scale systems. The analysis of ILSs in ROP_N becomes a new starting point of the future work. The colors are involved in the considered ROP_N models and used to regulate the routes of different work parts. In real systems, RFID chips or barcodes (QR codes) are used to regulate the routes of different work parts. It can be modeled by the presented SRS model. The work in this chapter also inspiring the study of ILS for the considering ROP_N, i.e., SRS, in the near future.

10.2 Future Work

The main open and interesting problems involved in ILS are twofold and briely outlined as follows.

10.2. FUTURE WORK

(I) Since a set of elementary siphons in the hybrid method is not unique, how to choose an optimal set of elementary siphons by considering their structures and initial markings is an important issue as the future work. In addition, application of the hybrid method should be extended to other classes of generalized Petri nets. This requires one to extend the concept of WSDCs and combine the mechanism behind ILSs with other siphon-based methods.

(II) The future work on LRS includes choosing an optimal set of competition paths to be controlled such that as many reachable states of the controlled system as possible can be obtained. Additionally, in order to avoid the NP-hard problem in mathematical programming, another way to compute control parameters needs to be considered by exploring the characteristic structure of WS^3PR . Extension of LRS to a more general class of Petri nets modeling more complex resource allocation systems remains an interesting widely open direction.

Furthermore, some classical mathematical problems can be further combined with the study of deadlock control in Petri net models. They are (1) the similar Chinese remainder problems, (2) the Knapsack problem, and (3) the Huarong Trail Problem. The first one is strongly connected with the special numerical relation in the generalized Petri nets. The second problem is a typical optimization problem, however, it will have a pre-condition, i.e., deadlock-freedom, when finding the optimal solution in the Petri net context. The last problem should be considered when the tokens in a net model have different sizes and the details of their moves in a resource place must be carefully controlled.

Besides, a special concern also needs to be addressed here. Mathematical programming becomes a prevailing method when analyzing and designing control policies for Petri net models. Lingo is one of the most powerful software packages for solving the various mathematical programming problems. However, sometime, rarely, it is not reliable.

As shown in Fig. 10.2, the software package is used to solve a Chinese remainder problem with the first 30 prime numbers except one. $@mod$ is a function to calculate the remainder. For example, $@mod(x, 5) \geq 3$ is a constraint and means that the remainder of x divided by 5 should be equal or greater than 3. The mathematical programming problem is to calculate the minimal x satisfying all 30 constraints. The version of Lingo is v12.0. The computer running Lingo is Think Station S3 with Intel Xeon CPU E5-1620, 3.60 GHz. After 40 minutes computation, it returns the

10.2. FUTURE WORK

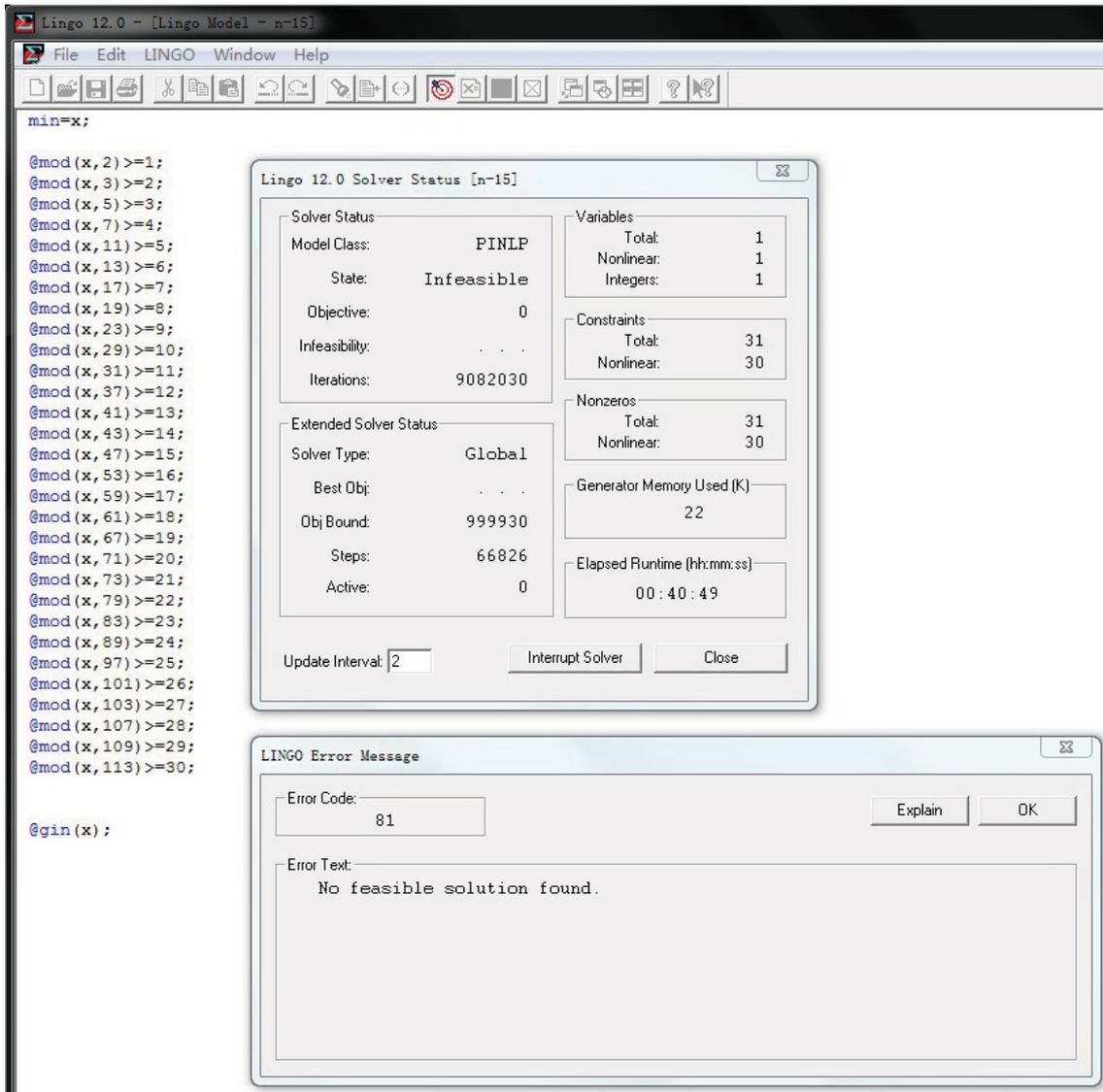


FIG. 10.2 – A wrong result given by Lingo.

10.2. FUTURE WORK

result showing that no feasible solution can be found, however, it is easy to find that 109169 is the solution. It implies that the dedicated software prototype should be developed for special problems instead of completely relying on the existing general software packages.

10.2. FUTURE WORK

Bibliographie

- [1] E. Badouel and P. Darondeau, “Theory of regions”, *Lectures on Petri Nets I : Basic Models*, LNCS, vol. 1491, pp. 529–586, 1998. 25
- [2] K. Barkaoui and J. F. Pradat-Peyre, “On liveness and controlled siphons in Petri nets,” in *Proceedings of the 17th International Conference on Applications and Theory of Petri Nets*, pp. 57–72, Osaka, Japan, Jun. 24-28, 1996. 24, 25
- [3] K. Barkaoui, A. Chaoui, and B. Zouari, “Supervisory control of discrete event systems based on structure theory of Petri nets,” in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, pp. 3750–3755, Orlando, USA, Oct. 12-15, 1997. 25
- [4] K. Barkaoui, J. M. Couvreur, and K. Klai, “On the equivalence between liveness and deadlock-freeness in Petri nets,” in *Proceedings of the 26th International Conference on Application and Theory of Petri Nets*, pp. 90–107, Miami, USA, Jun. 20-24, 2005. 25
- [5] P. Buchholz, “Hierarchical high level Petri nets for complex system analysis,” in *Proceedings of the 15th International Conference on Application and Theory of Petri Nets*, pp. 119–138, Zaragoza, Spain, Jun. 20-24, 1994. 114
- [6] P. Buchholz and P. Kemper, “Hierarchical reachability graph generation for Petri nets,” *Formal Methods in System Design*, vol. 21, no. 3, pp. 281–315, 2002. 114
- [7] CPN Tools Homepage, <http://cpntools.org/> 18, 160
- [8] J. Carmona, J. Cortadella, and M. Kishinevsky, “Divide-and-conquer strategies for process mining,” in *Proceedings of Business Process Management*, pp. 327–343, Ulm, Germany, Sep. 8-10, 2009. 114

BIBLIOGRAPHIE

- [9] D. Y. Chao, "Max'-controlled siphons for liveness of S^3PGR^2 ," *IET Control Theory and Applications*, vol. 1, no. 4, pp. 933–936, 2007. 24, 25
- [10] D. Y. Chao, "Direct minimal empty siphon computation using MIP," *International Journal of Advanced Manufacturing Technology*, vol. 45, no. 3-4, pp. 397–405, 2009. 133
- [11] Y. F. Chen, Z. W. Li, M. Khalgui, and O. Mosbahi, "Design of a maximally permissive liveness-enforcing Petri net supervisor for flexible manufacturing systems," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 2, pp. 374–393, 2011. 25, 75
- [12] Y. F. Chen and Z. W. Li, "Design of a maximally permissive liveness-enforcing supervisor with a compressed supervisory structure for flexible manufacturing systems," *Automatica*, vol. 47, no. 5, pp. 1028–1034, 2011. 25
- [13] H. Cho, T. K. Kumaran, and R. A. Wysk, "Graph-theoretic deadlock detection and resolution for flexible manufacturing systems," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 3, pp. 413–421, 1995. 23, 25
- [14] F. Chu and X. L. Xie, "Deadlock analysis of Petri nets using siphons and mathematical programming," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 6, pp. 793–804, 1997. 133
- [15] E. G. Coffman, M. J. Elphick, and A. Shoshani, "System deadlocks," *ACM Computing Surveys*, vol. 3, no. 2, pp. 67–78, 1971. 23
- [16] J. Ezpeleta, J. M. Colom, J. and Martínez, "A Petri net based deadlock prevention policy for flexible manufacturing systems," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 2, pp. 173–184, 1995. 24, 25, 70, 75, 94, 133, 145
- [17] J. Ezpeleta, F. García-Vallés, J. M. and Colom, "A class of well structured Petri nets for flexible manufacturing systems," in *Proceedings of the 19th International Conference on Application and Theory of Petri Nets*, pp. 64–83, Lisbon, Portugal, Jun. 22-26, 1998. 24
- [18] M. P. Fanti, B. Maione, and B. Turchiano, "Comparing digraph and Petri net approaches to deadlock avoidance in FMS," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 30, no. 5, pp. 783–798, 2000. 23

- [19] M. P. Fanti and M. C. Zhou, "Deadlock control methods in automated manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part A : Systems and Humans*, vol. 34, no. 1, pp. 5–22, 2004. 23
- [20] J. Freiheit and A. Zimmermann, "A divide and conquer approach for the performance evaluation of large stochastic Petri nets," in *Proceedings of the 9th International Workshop on Petri Nets and Performance Models*, pp. 91–100, Aachen, Germany, Sep. 11-14, 2001. 114
- [21] M. M. Gao, M. C. Zhou, and R. J. Caudill, "Integratration of disassembly leveling and bin assignment for demanufacturing automation," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 6, pp. 867–874, 2002.
- [22] M. M. Gao, M. C. Zhou, and Y. Tang, "Intelligent decision making in disassembly process based on fuzzy reasoning Petri nets," *IEEE Transactions on Systems, Man, and Cybernetics, Part B : Cybernetics*, vol. 34, no. 5, pp. 2029–2034, 2004.
- [23] A. Ghaffari, N. Rezg, and X. L. Xie, "Design of a live and maximally permissive Petri net controller using the theory of regions," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 137–142, 2003. 25, 70, 74
- [24] C. Girault and R. Valk, *Petri Nets for Systems Engineering - A Guide to Modeling, Verification, and Applications*, Berlin, Germany, Springer-Verlag, 2003. 75
- [25] A. Gungor and S. M. Gupta, "Issues in enviromentally conscious manufacuting and product recovery : a survery," *Computers and Industrial Engineering*, vol. 36, pp. 811–853, 1996.
- [26] O. Gunther, W. Kletti, and U. Kubach, *RFID in Manufacturing*, Berlin : Springer, 2008.
- [27] Y. F. Hou, D. Liu, Z. W. Li, and M. Zhao, "Deadlock prevention using divide-and-conquer strategy for WS³PR," in *Proceedings of IEEE International Conference on Mechatronics and Automation*, pp. 1635–1640, Xi'an, China, Aug. 4-7, 2010. 115, 117, 120, 128
- [28] Y. F. Hou and Z. W. Li, "Extended elementary siphon-based deadlock prevention policy for flexible manufacturing systems," submitted to *Asian Journal of Control*, 2012. 151
- [29] B. Hruz and M. C. Zhou, *Modeling and Control of Discrete-event Dynamic Systems with Petri Nets and Other Tool*, London : Springer, 2007.

BIBLIOGRAPHIE

- [30] F. S. Hsieh, "Robustness analysis of non-ordinary Petri nets for flexible assembly/disassembly processes based on structural decomposition," *International Journal of Control*, vol. 84, no. 3, pp. 496–510, 2011.
- [31] F. S. Hsieh, "Robustness analysis of non-ordinary Petri nets for flexible assembly systems," *International Journal of Control*, vol. 83, no. 5, pp. 928–939, 2010.
- [32] H. S. Hu, M. C. Zhou, and Z. W. Li, "A new class of Petri nets for modeling and control of ratio-enforced resource allocation systems," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, pp. 199–204, San Antonio, USA, Oct. 11-14, 2009. 108
- [33] H. S. Hu, M. C. Zhou, and Z. W. Li, "Liveness enforcing supervision of video streaming systems using non-sequential Petri nets," *IEEE Transactions on Multimedia*, vol. 11, no. 8, pp. 1446–1456, 2009.
- [34] H. S. Hu, M. C. Zhou, and Z. W. Li, "Low-cost and high-performance supervision in ratio-enforced automated manufacturing systems using timed Petri nets," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 4, pp. 933–944, 2010. 133
- [35] H. S. Hu and Z. W. Li, "Synthesis of liveness enforcing supervisor for automated manufacturing systems using insufficiently marked siphons," *Journal of Intelligent Manufacturing*, vol. 21, no. 4, pp. 555–567, 2010.
- [36] H. S. Hu, M. C. Zhou, and Z. W. Li, "Supervisor design to enforce production ratio and absence of deadlock in automated," *IEEE Transactions on Systems, Man, and Cybernetics, Part A : Systems and Humans*, vol. 41, no. 2, pp. 201–212, 2011. 108
- [37] Y. S. Huang, M. D. Jeng, X. L. Xie, and S. L. Chung, "A deadlock prevention policy for flexible manufacturing systems using siphons," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 541–546, Seoul, Korea, May 21-26, 2001. 24, 75, 133, 142
- [38] Y. S. Huang, M. D. Jeng, X. L. Xie, and S. L. Chung, "Deadlock prevention policy based on Petri nets and siphons," *International Journal of Production Research*, vol. 39, no. 2, pp. 283–305, 2001.

- [39] Y. S. Huang, M. D. Jeng, X. L. Xie, and D. H. Chung, "Siphon-based deadlock prevention policy for flexible manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part A : Systems and Humans*, vol. 36, no. 6, pp. 2152–2160, 2006.
- [40] Y. S. Huang, "Design of deadlock prevention supervisors using Petri nets," *International Journal of Advanced Manufacturing Technology*, vol. 35, no. 3-4, pp. 349–362, 2007.
- [41] M. D. Jeng and F. DiCesare, "Synthesis using resource control nets for modeling shared-resource systems," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 3, pp. 317–327, 1995.
- [42] K. Jensen and L. M. Kristensen, *Coloured Petri Nets Modeling and Validation of Concurrent Systems*, London : Springer, 2009.
- [43] D. B. Johnson, "Finding all the elementary circuits of a directed graph," *SIAM Journal of Computing*, vol. 4, no. 1, pp. 77–84, 1975. 135, 148
- [44] S. Kalpakjian and S. Schmid, *Manufacturing Engineering and Technology*, Prentice Hall, 2009.
- [45] D. Karlsson, P. Eles, Z. Peng, "Formal verification of component-based designs," *Design Automation for Embedded Systems*, vol. 11, no. 1, pp. 49–90, 2007. 114
- [46] K. Klai, L. Petrucci, and M. Reniers, "An incremental and modular technique for checking LTL\X properties of Petri nets," in *Proceedings of 27th International Conference on Formal Techniques for Networked and Distributed Systems*, pp. 280–295, Tallinn, Estonia, Jun. 27–29, 2007. 114
- [47] I. Koh and F. DiCesare, "Modular transformation methods for generalized Petri nets and their application to automated manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 6, pp. 1512–1522, 1991. 108
- [48] D. E. Knuth, *The Art of Computer Programming : Volumn 3 Sorting and Searching*, Addison-Wesley, 1998. 113
- [49] A. J. D. Lambert, "Disassembly sequencing : a survey", *International Journal of Production Research*, vol. 41, no. 16, pp. 3721–3759, 2003.

BIBLIOGRAPHIE

- [50] J. K. Lee and O. Korbaa, "Modeling and scheduling of ratio-driven FMS using unfolding time Petri nets," *Computers & Industrial Engineering*, vol. 46, no. 4, pp. 639–653, 2004.
- [51] J. S. Lee and P. L. Hsu, "A new approach to evaluate ladder logic diagrams and Petri nets via the if-then transformation," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, pp. 2711–2716, Tucson, USA, Oct. 7-10, 2001.
- [52] F. L. Lewis, A. Gürel, S. Bogdan, A. Doğanalp, and O. C. Pastravanu, "Analysis of deadlock and circular waits using a matrix model for flexible manufacturing systems," *Automatica*, vol. 34, no. 9, pp. 1083–1100, 1998. 25, 44, 46, 47, 51, 69, 75, 87, 141
- [53] Z. W. Li and M. C. Zhou, "Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part A : Systems and Humans*, vol. 34, no. 1, pp. 38–51, 2004. 74, 75, 142, 146
- [54] Z. W. Li and M. C. Zhou, "Clarifications on the Definitions of Elementary Siphons of Petri Nets," *IEEE Transactions on Systems, Man, and Cybernetics, Part A : Systems and Humans*, vol. 36, no. 6, pp. 1227–1229, 2006. 74, 75
- [55] Z. W. Li and M. C. Zhou, "Two-stage method for synthesizing liveness-enforcing supervisors for flexible manufacturing systems using Petri nets," *IEEE Transactions on Industrial Informatics*, vol. 2, no. 4, pp. 313–325, 2006. 89
- [56] Z. W. Li and D. Liu, "A correct minimal siphons extraction algorithm from a maximal unmarked siphon of a Petri net," *International Journal of Production Research*, vol. 45, no. 9, pp. 2163–2167, 2007.
- [57] Z. W. Li, H. S. Hu, and A. R. Wang, "Design of liveness-enforcing supervisors for flexible manufacturing systems using Petri nets," *IEEE Transactions on Systems, Man, and Cybernetics, Part C : Applications and Reviews*, vol. 37, no. 4, pp. 517–526, 2007.
- [58] Z. W. Li and M. Zhao, "On controllability of dependent siphons for deadlock prevention in generalized Petri nets," *IEEE Transactions on Systems, Man, and Cybernetics, Part A : Systems and Humans*, vol. 38, no. 2, pp. 369–384, 2008. 25, 70, 74, 75, 88, 94, 133

- [59] Z. W. Li and M. C. Zhou, "On siphon computation and deadlock control for a class of Petri nets," *IEEE Transactions on Systems, Man, and Cybernetics, Part A : Systems and Humans*, vol. 38, no. 3, pp. 667–679, 2008. 74, 75
- [60] Z. W. Li, M. C. Zhou, and N. Q. Wu, "A survey and comparison of Petri net-based deadlock prevention policies for flexible manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C : Applications and Reviews*, vol. 38, no. 2, pp. 173–188, 2008. 75, 89
- [61] Z. W. Li and M. C. Zhou, *Deadlock Resolution in Automated Manufacturing Systems A Novel Petri Net Approach*, Springer, London, 2009. 25, 33, 74, 75, 94
- [62] Z. W. Li, S. Zhu, and M. C. Zhou, "A divide-and-conquer strategy to deadlock prevention in flexible manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C : Applications and Reviews*, vol. 39, no. 2, pp. 156–169, 2009. 89, 114, 117, 119, 120, 127, 128
- [63] Ding Liu, Kamel Barkaoui, and MengChu Zhou, "On intrinsically live structure of a class of generalized Petri nets modeling FMS," in *Proceedings of 11th International Workshop on Discrete Event Systems*, pp. 187–192, Guadalajara, Mexico, Oct. 3-5, 2012.
- [64] D. Liu, Z. W. Li, and M. C. Zhou, "Liveness of an extended S^3PR ," *Automatica*, vol. 46, no. 6, pp. 1008–1018, 2010. 25, 26, 55, 58, 87, 145
- [65] D. Liu, Z. W. Li, and M. C. Zhou, "Erratum to "Liveness of an extended S^3PR [Automatica 46 (2010) 1008-1018]"", *Automatica*, vol. 48, no. 5, pp. 1003–1004, 2012. 25, 33
- [66] D. Liu, Z. W. Li, and M. C. Zhou, "Hybrid liveness-enforcing policy for generalized Petri net models of flexible manufacturing systems," accepted by *IEEE Transactions on Systems, Man, and Cybernetics, Part A : Systems and Humans*, 2012, DOI : 10.1109/TSMCA.2012.2192266. 25, 33
- [67] D. Liu, Z. W. Li, and M. C. Zhou, "A parameterized liveness and ratio-enforcing supervisor for a class of generalized Petri nets," *Automatica*, vol. 49, no. 11, pp.3167–3179, 2013.
- [68] R. M. Mahoney, *High-Mix-Low-Volume Manufacturing*, Prentice Hall PTR, 1997. 23

BIBLIOGRAPHIE

- [69] O. Marchetti and A. Munier-Kordon, "A sufficient condition for the liveness of weighted event graphs," *European Journal of Operational Research*, vol. 197, no. 2, pp. 532–540, 2009.
- [70] O. Marchetti and A. Munier-Kordon, "Complexity results for weighted timed event graphs," *Discrete Optimization*, vol. 7, no. 3, pp. 166–180, 2010.
- [71] V. Mertsiotakis and M. Silva, "Throughput approximation of decision free processes using decomposition," in *Proceedings of the 7th International Workshop on Petri Nets and Performance Models*, pp. 174–182, St. Malo, France, Jun. 3-6, 1997. 114
- [72] T. Murata, "Petri nets : properties, analysis and applications," *Proceedings of The IEEE*, vol. 77, no. 4, pp. 541–580, 1989. 33
- [73] A. Nazeem, S. Reveliotis, Y. Wang, and S. Lafortune, "Designing compact and maximally permissive deadlock avoidance policies for complex resource allocation systems through classification theory : the linear case," *IEEE Transactions on Automatic Control*, vol. 56, no. 8, pp 1818–1833, 2011. 133
- [74] O. Oanea, H. Wimmel, and K. Wolf, "New algorithms for deciding the siphon-trap property," in *Proceedings of the 31st International Conference on Applications and Theory of Petri Nets and Other Models of Concurrency*, pp. 267–286, Braga, Portugal, Jun. 21-25, 2010. 114
- [75] H. Ohl, H. Camus, E. Castelain, and J. C. Gentina, "Petri net modeling of ratio-driven flexible manufacturing systems and implications on the WIP for cyclic schedules," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, pp. 3081–3086, Vancouver, Canada, Oct. 22-25, 1995. 108
- [76] A. Ohta and A. Tsuji, "Insufficiently marked siphon of Petri nets-extension of token-free siphon," in *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. III244–III247, Bangkok, Thailand, May 25-28, 2003.
- [77] J. Park, *Structural Analysis and Control of Resource Allocation Systems Using Petri Nets*, PhD Dissertation, Georgia Institute of Technologe, Atlanta, USA, 2000.

BIBLIOGRAPHIE

- [78] J. Park and S. A. Reveliotis, "Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings," *IEEE Transactions on Automatic Control*, vol. 46 no. 10, pp. 1572–1583, 2001. 24, 25, 70
- [79] C. A. Petri and W. Reisig, "Petri nets," *Scholarpedia*, 3(4) : 6477, http://www.scholarpedia.org/article/Petri_net, 2008. 23
- [80] Petri Nets World, <http://www.informatik.uni-hamburg.de/TGI/PetriNets/>
- [81] P. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM Journal on Control and Optimization*, vol. 25, no. 1, pp. 206–230, 1987. 23, 25
- [82] S. Reveliotis, "Implicit siphon control and its role in the liveness-enforcing supervision of sequential resource allocation systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part A : Systems and Humans*, vol. 37, no. 3, pp. 319–328, 2007. 23
- [83] A. Santoyo-Sanchez, A. Ramire-Trevintildeo, C. De Jesus-Velasquez, and L. I. Aguirre-Salas, "Step state-feedback supervisory control of discrete event systems using interpreted Petri nets," in *Proceedings of the 13th IEEE International Conference on Emergine Technologies and Factory Automation*, pp. 926–933, Hamburg, Germany, Sep. 15-18, 2008. 114
- [84] H. I. Son, "Design and implementation of decentralised supervisory control for manufacturing system automation," *International Journal of Computer Integrated Manufacturing*, vol. 24, no. 3, pp. 242–256, 2011. 114
- [85] Y. J. Song and J. K. Lee, "Analysis of Petri net models using transitive matrix," in *Proceedings of IEEE Internaional Conference on Systems, Man, and Cybernetics*, pp. 3122–3127, Nashville, USA, Oct. 8-11, 2000. 114
- [86] I. Suzuki and T. Murata, "A method for stepwise refinement and abstraction of Petri nets," *Journal of Computer and System Sciences*, vol. 27, no. 1, pp. 51–76, 1983. 114
- [87] J. L. Szwarcfiter and P. E. Lauer, "A search strategy for the elementary cycles of a directed graph," *BIT*, vol. 16. no. 2, pp. 192–204, 1976.

BIBLIOGRAPHIE

- [88] Y. Tang, M. C. Zhou, and R. G. Qiu, "Virtual production lines design for back-end semiconductor manufacturing systems," *IEEE Transactions on Semiconductor Manufacturing*, vol. 16, no. 3, pp. 543–550, 2003.
- [89] Y. Tang, M. C. Zhou, and M. M. Gao, "Fuzzy-Petri-net-based disassembly planning considering human factors," *IEEE Transactions on Systems, Man, and Cybernetics, Part A : Systems and Humans*, vol. 36, no. 4, pp. 718–726, 2006.
- [90] R. Tarjan, "Enumeration of the elementary circuits of a directed graph," *SIAM Journal on Computing*, vol. 2, no. 3, pp. 211–216, 1973.
- [91] F. Tricas, and J. Martínez, "An extension of the liveness theory for concurrent sequential processes competing for shared resources," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, pp. 3035–3040, Vancouver, USA, Oct. 22-25, 1995. 25, 69
- [92] F. Tricas, F. García-Vallés, J. M. Colom, and J. Ezpeleta, "A structure approach to the problem of deadlock prevention in process with shared resources," in *Proceedings of the 4th Workshop on Discrete Evnet Systems*, pp. 273–278, Cagliari, Italy, Aug. 26-28, 1998. 142, 145
- [93] F. Tricas, F. García-Vallés, J. M. Colom, and J. Ezpeleta, "An iterative method for deadlock prevention in FMS," in *Proceedings of the 5th Workshop on Discrete Event Systems*, pp. 139–148, Ghent, Belgium, Aug. 21-23, 2000. 25
- [94] F. Tricas, *Deadlock Analysis, Prevention and Avoidance in Sequential Resource Allocation Systems*, PhD Dissertation, University of Zaragoza, Spain, 2003. 23
- [98] M. Uzam, "An optimal deadlock prevention policy for flexible manufacturing systems using Petri net models with resources and the theory of regions," *International Journal of Advanced Manufacturing Technology*, vol. 19, no. 3, pp. 192–208, 2002. 25, 70, 74
- [96] M. Uzam and M. C. Zhou, "An improved iterative synthesis method for liveness enforcing supervisors of flexible manufacturing systems," *International Journal of Production Research*, vol. 44, no. 10, pp. 1987–2030, 2006.

BIBLIOGRAPHIE

- [97] M. Uzam and M. C. Zhou, "An iterative synthesis approach to Petri net based deadlock prevention policy for flexible manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part A : Systems and Humans*, vol. 37, no. 3, pp. 362–371, 2007. 25
- [98] M. Uzam, "An optimal deadlock prevention policy for flexible manufacturing systems using Petri net models with resources and the theory of regions," *International Journal of Advanced Manufacturing Technology*, vol. 19, no. 3, pp. 192–208, 2002. 25, 70, 74
- [99] W. van der Aalst, A. ter Hofstede, B. Kiepuszewski, and A. Barros, "Workflow pattern," *Distributed and Parallel Databases*, vol. 14, no. 3, pp. 5–51, 2003.
- [100] P. Veerakamolmal and S. M. Gupta, "High-mix/low-volume batch of electronic equipment disassembly," *Computers and Industrial Engineering*, vol. 35, no. 1-2, pp. 65-68, 1998.
- [101] K. Venkatesh, M. C. Zhou, and R. J. Caudill, "Comparing ladder logic diagrams and Petri nets for sequence controller design through a discrete manufacturing systems," *IEEE Transactions on Industrial Electronics*, vol. 41, no. 6, pp. 611-619, 1994.
- [102] J. C. Wang, *Timed Petri Nets : Theory and Application*, Boston : Kluwer Academic Publishers, 1998.
- [103] A. R. Wang, Z. W. Li, J. Y. Jia, and M. C. Zhou, "An efficient algorithm to find elementary siphons in a class of Petri nets," *IEEE Transactions on Systems, Man, and Cybernetics, Part A : Systems and Humans*, vol. 39, no. 4, pp. 912–923, 2009.
- [104] W. M. Wonham and P. J. Ramadeg, "Modular supervisory control of discrete-event systems," *Mathematics of Control, Signals, and Systems*, vol. 1, no. 1, pp. 13–30, 1988. 23
- [105] W. M. Wonham, *Supervisory Control of Discrete-Event Systems*, <http://www.control.utoronto.ca/cgi-bin/dldes.cgi>, 2011.
- [106] N. Q. Wu, "Necessary and sufficient conditions for deadlock-free operation in flexible manufacturing systems using a colored Petri net model," *IEEE Transactions on Systems, Man, and Cybernetics, Part C : Applications and Reviews*, vol. 29, no. 2, pp. 192–204, 1999. 24

BIBLIOGRAPHIE

- [107] N. Q. Wu and M. C. Zhou, "Avoiding deadlock and reducing starvation and blocking in automated manufacturing systems," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 5, pp.657–668, 2001.
- [108] N. Q. Wu and M. C. Zhou, "Modeling and deadlock control of automated guided vehicle systems," *IEEE Transactions on Mechatronics*, vol. 9, no. 1, pp. 50–57, 2004.
- [109] N. Q. Wu and M. C. Zhou, "Modelling and deadlock avoidance of automated manufacturing systems with multiple automated guided vehicles," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 35, no. 6, pp. 1193–1202, 2005.
- [110] N. Q. Wu and M. C. Zhou, "Shortest routing of bi-directional automated guided vehicles avoiding deadlock and blocking," *IEEE Transactions on Mechatronics*, vol. 12, no. 1, pp. 63–72, 2007.
- [111] N. Q. Wu, M. C. Zhou, and Z. W. Li, "Resource-oriented Petri net for deadlock avoidance in flexible assembly systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part A : Systems and Humans*, vol. 38, no. 1, pp. 56–69, 2008.
- [112] N. Q. Wu and M. C. Zhou, *System Modeling and Control with Resource-Oriented Petri Nets*, CRC Press, 2010. 24
- [113] N. Q. Wu and M. C. Zhou, "Process vs resource-oriented Petri net modeling of automated manufacturing systems," *Asian Journal of Control*, vol. 12, no. 3, pp. 267–280, 2010.
- [114] N. Q. Wu and M. C. Zhou, "Intelligent token Petri nets for modelling and control reconfigurable automated manufacturing systems with dynamical changes," *Transactions of the Institute of Measurement and Control*, vol. 33, no. 1, pp. 9–29, 2011. 157
- [115] S. Y. Yan, *Number Theory for Computing (2nd Edition)*, Springer, Berlin, 2002. 79
- [116] R. A. Wysk, N. S. Yang, and S. Joshi, "Detection of deadlocks in flexible manufacturing cells," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 6, pp. 853–859, 1991.
- [117] K. Y. Xing, B. S. Hu, and H. X. Chen, "Deadlock avoidance policy for Petri net modeling of flexible manufacturing systems with shared resources," *IEEE Transactions on Automatic Control*, vol. 41, no. 2, pp. 289–295, 1996.

BIBLIOGRAPHIE

- [118] K. Y. Xing and B. S. Hu, "Optimal liveness Petri net controllers with minimal structures for automated manufacturing systems," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, pp. 282–287, Hawaii, USA, Oct. 10-12, 2005.
- [119] K. Y. Xing, M. C. Zhou, X. Yang, and F. Tian, "Optimal Petri net-based polynomial-complexity deadlock avoidance policies for automated manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part A : Systems and Humans*, vol. 39, no. 1, pp. 188-199, 2009.
- [120] M. Zhao and Z. W. Li, "On deadlock control for a class of generalized Petri net models of flexible manufacturing systems," *International Journal of Production Research*, vol. 47, no. 21, pp. 6085–6107, 2009. 13, 151, 152
- [121] M. Zhao, Y. F. Hou, and D. Liu, "Liveness-enforcing supervisors synthesis for a class of generalized Petri nets based on two-stage deadlock control and mathematical programming," *International Journal of Control*, vol. 83, no. 10, pp. 2053–2066, 2010. 133, 136
- [122] C. F. Zhong and Z. W. Li, "Design of liveness-enforcing supervisors via transforming plant Petri net models of FMS," *Asian Journal of Control*, vol. 12, no. 3, pp. 240–252, 2010. 25
- [123] C. F. Zhong and Z. W. Li, "Self-liveness of a class of Petri net models for flexible manufacturing systems," *IET Control Theory and Applications*, vol. 4, no. 3, pp. 403–410, 2010.
- [124] M. C. Zhou and F. DiCesare, "Parallel and sequential exclusions for Petri net modeling for manufacturing systems," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 4, pp. 515–527, 1991. 70
- [125] M. C. Zhou, F. DiCesare, and A. A. Desrochers, "A hybrid methodology for synthesis of Petri nets for manufacturing systems," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 350–361, 1992. 70
- [126] M. C. Zhou, F. DiCesare, and D. Rudolph, "Design and implementation of a Petri net based supervisor for a flexible manufacturing system," *Automatica*, vol. 28, no. 6, pp. 1199–1208, 1992.

BIBLIOGRAPHIE

- [127] M. C. Zhou and F. DiCesare, *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*, Kluwer, London, 1993. 23, 24
- [128] M. C. Zhou and K. Venkatesh, *Simulation and Control of Flexible Manufacturing Systems, A Petri Net Approach*, World Scientific, Singapore, 1998. 23
- [129] M. C. Zhou and M. P. Fanti, *Deadlock Resolution in Computer-integrated Systems*, New York : Marcel Dekker, 2005.
- [130] B. Zouari and K. Barkaoui, "Parameterized Supervisor Synthesis for a Modular Class of Discrete Event Systems," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1874-1879, Washington, USA, Oct. 5-8, 2003. 24, 25
- [131] R. Zurawski and M. C. Zhou, "Petri nets and industrial applications : a tutorial," *IEEE Transactions on Industrial Electronics*, vol. 41, no. 6, pp. 567–583, 1994.

Glossaire

- CB : circular blocking
- CIM : computer integrated manufacturing
- CW : circular wait
- D&C : divide-and-conquer
- DES : discrete event system
- ES³PR : extended system of simple sequential processes with resources
- FMS : flexible manufacturing system
- GS³PR : general system of simple sequential processes with resources
- ILS : intrinsically live structure
- LRS : liveness and ratio-enforcing supervisor
- MIMS : maximal insufficiently marked siphon
- MIP : mixed integer programming
- POPN : process-oriented Petri net
- RAS : resource allocation system
- ROPN : resource-oriented Petri net
- RU-ratio : resource usage ratio
- S²P : simple sequential processes
- S²PR : simple sequential processes with resources
- S³PR : system of simple sequential processes with resources
- S³PMR : system of simple sequential processes with multiple resources
- S³PGR² : system of simple sequential processes with general resource requirements
- SCT : supervisory control theory
- SMS : strict minimal siphon

- S^4R : system of sequential systems with shared resources
- WS^2PR : simple sequential process with weighted resource allocation
- WS^3PR : system of simple sequential processes with weighted resource allocation
- WS^3PSR : weighted system of simple sequential processes with several resources
- $WSDC$: weighted simple directed circuit

Résumé en Français

On donne ici chapitre par chapitre une description simplifiée de la thèse en utilisant les figures, exemples et évitant les équations mathématiques. L'objet ici est d'essayer d'introduire l'essence des méthodes proposées avec langage naturel.

Le chapitre 1 motive et présente le sujet. Il donne un état de l'art concis de l'analyse structurelle des réseaux de Petri pour l'étude du blocage et de la vivacité.

Ce chapitre se termine par un résumé de la thèse dégageant parfaitement les contributions décrites dans la thèse.



Fig. 2 The two-oxen deadlock sometimes happens in real world, such as traffic and manufacturing systems.

La situation d'interblocage est observée dans le monde réel et en particulier dans le domaine des systèmes d'allocation de ressources tels que les le systèmes de production flexible (FMS). Elle signifie la compétition de deux processus concurrents sur une même capacité de ressource d'un système. De nombreux travaux ont traite de l'analyse et du contrôle de cette situation. Qu'en est il si deux travaux distincts exigent différentes capacités d'une ressource , par exemple $2/3$ et $1/2$ ou $2/3$ et $1/3$ tel que décrit dans Fig. 3. Peu de travaux dans la littérature ont traite ce cas.

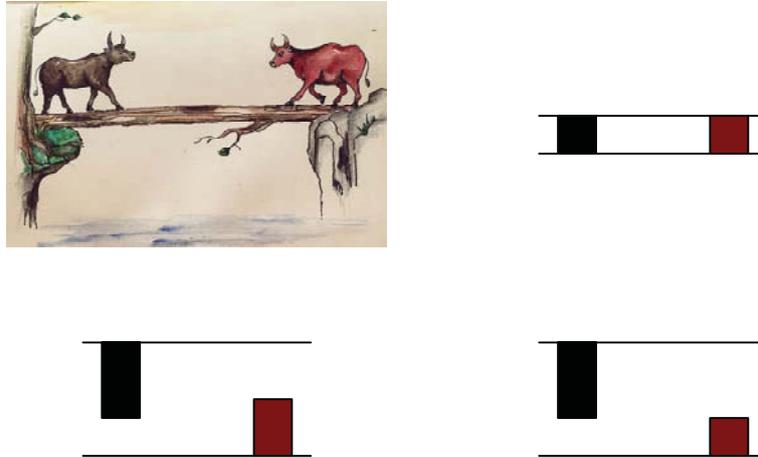


Fig. 3 A two-oxen deadlock can be abstract a simple model in which two concurrent jobs require the same capacity of a system resource. What if the two jobs require different capacities of the system resource, like the 2/3 and 1/2 situation or the 2/3 and 1/3 one?

Le cas de la Figure 4 rend plus complexe le contrôle de l'interblocage

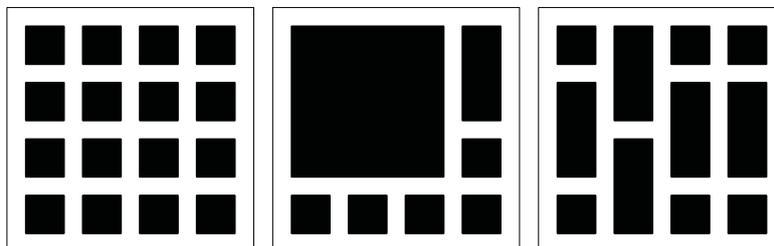


Fig. 4 If all jobs in a system require the same capacity of a system resource. It makes the deadlock control relatively easy, whereas if different jobs require different capacities of the system resource, the deadlock control becomes complex.

Dans ce travail, nous utilisons le formalisme des réseaux de Petri pour analyser et raisonner sur ce problème. Les travaux sur les systèmes concurrents basés sur la théorie et l'application des réseaux de Petri ont connu un grand développement ces dernières années (Voir l'histogramme de la T fig. 5 extrait de Google Scholar).

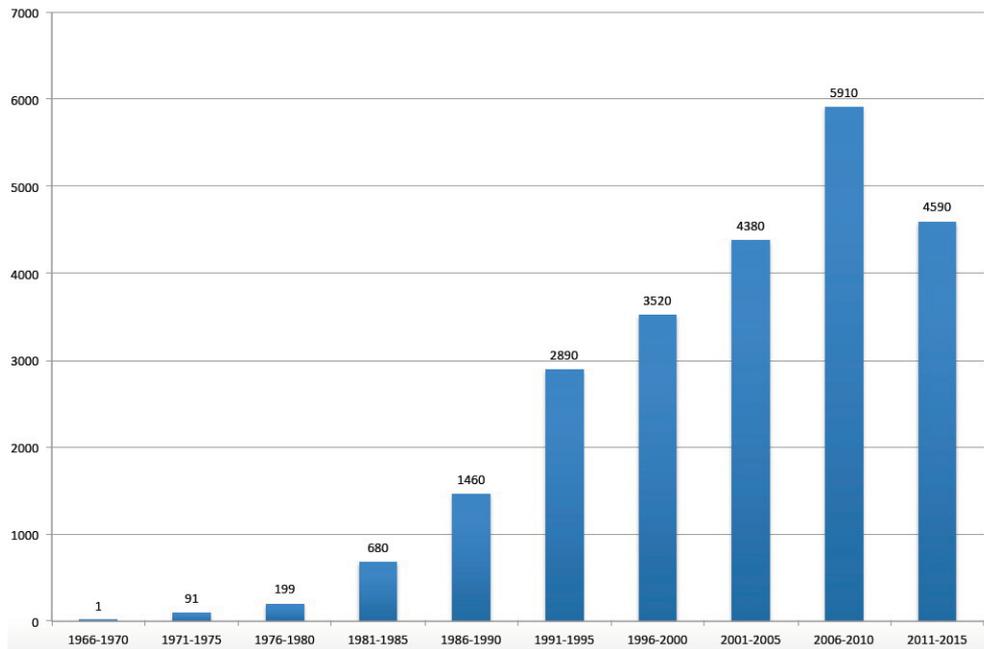


Fig. 5 A histogram

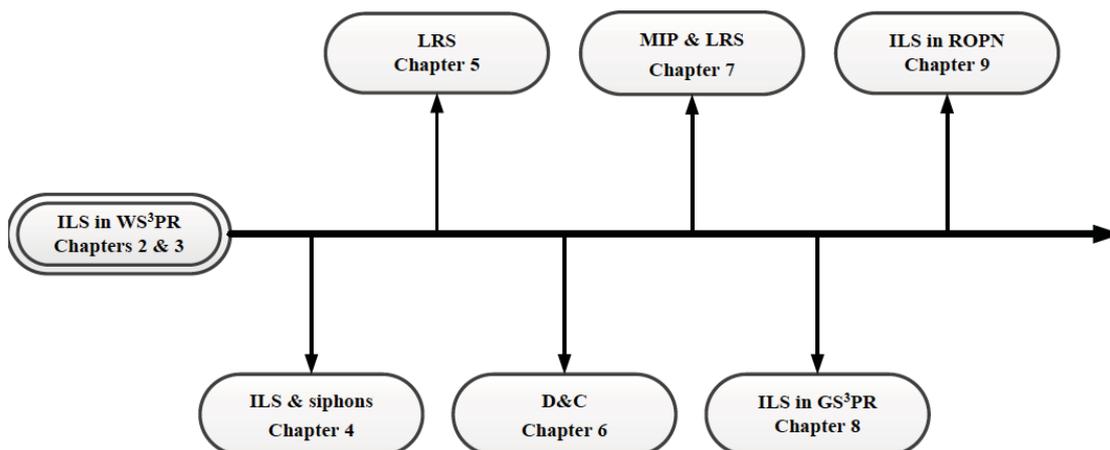


Fig. 6 L'organisation du travail de la thèse

Comme indiqué dans la Fig. 6, le travail présenté dans cette thèse est fortement lié à l'étude de la structure intrinsèquement vivantes (intrinsically live structures ou ILS) dans la classe des RdP généralisés appelée WS3PR.

Le chapitre 2 est un chapitre court fixant les notations nécessaires à la présentation relatives aux réseaux de Petri.

Comme le montre la Fig. 7, lorsque le poids des arcs dans les RdP est supérieur ou égal à 1, le contrôle des interblocages devient une tâche très difficile.

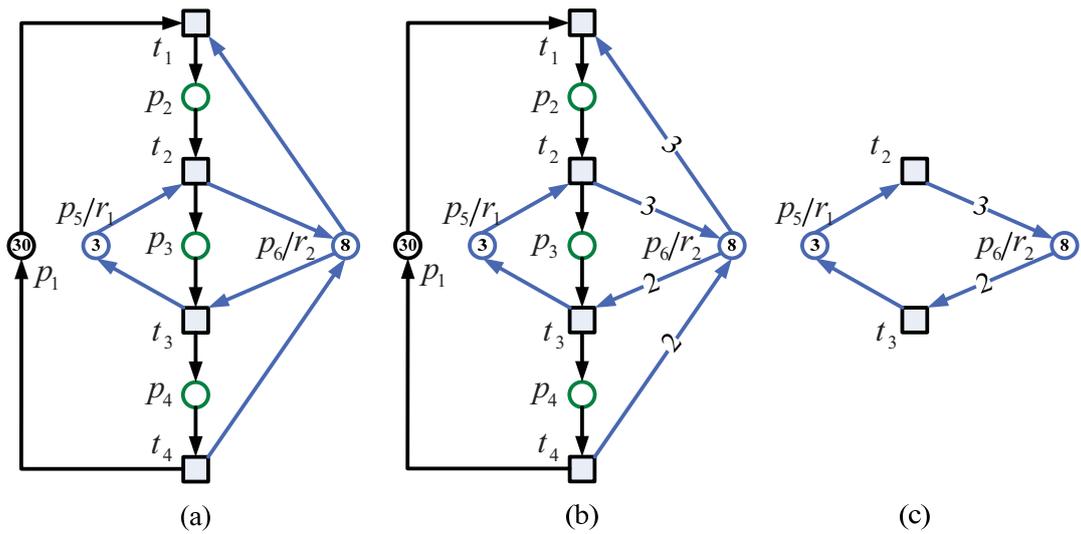


Fig. 7 (a) An ordinary net, (b) a generalized net, and (c) a weighted digraph of resource places

Comme la montre la Fig. 8 (a), la classe des WS3PR est une extension des S3PR qui sont largement utilisés pour modéliser les FMS. Dans ce travail on se focalise particulièrement sur le sous réseau induit par les places ressources (voir Fig. 8(b)).

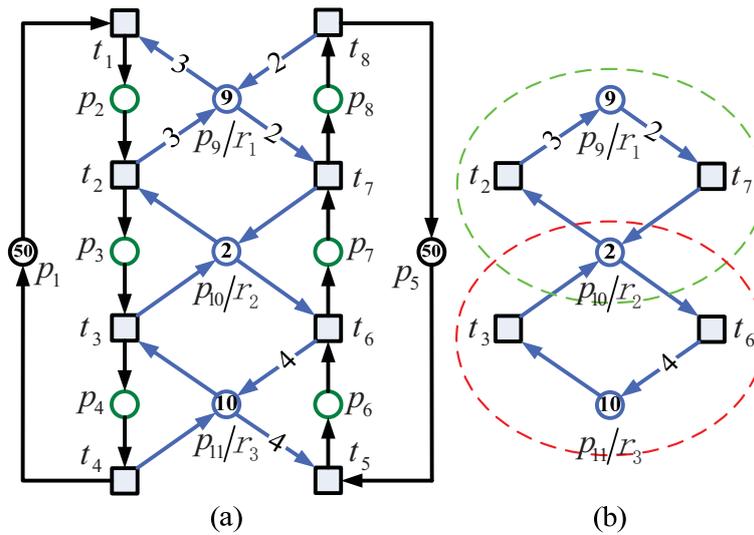


Fig. 8 (a) A generalized net and (b) one of its subnet

Le chapitre 3 introduit le modèle des WS3PR (System of Simple Sequential Processes with Weighted Resource Allocation). On établit une caractérisation fine de la vivacité d'un tel modèle. Elle repose sur la notion d'attente circulaire. Une attente circulaire peut être vue comme une composante connexe du sous graphe réduit aux transitions et aux places ressources du modèle.

Nous pouvons remarquer que les transitions de l'attente circulaire restent bloquées à jamais si d'une part, les places ressources de l'attente sont insuffisamment marquées et que les places de

contrôle partageant les ressources de l'attente et ne partageant pas de transitions en entrée sont vides. On introduit ainsi la notion de blocage circulaire comme un tel marquage amenant au blocage des transitions d'une attente circulaire.

Enfin, on démontre un théorème essentiel à notre démarche : la non vivacité d'un WS3PR est équivalente à l'existence d' « un blocage circulaire dans une attente circulaire ».

Ce résultat est très intéressant et généralise finement la caractérisation de la vivacité d'un S3PR.

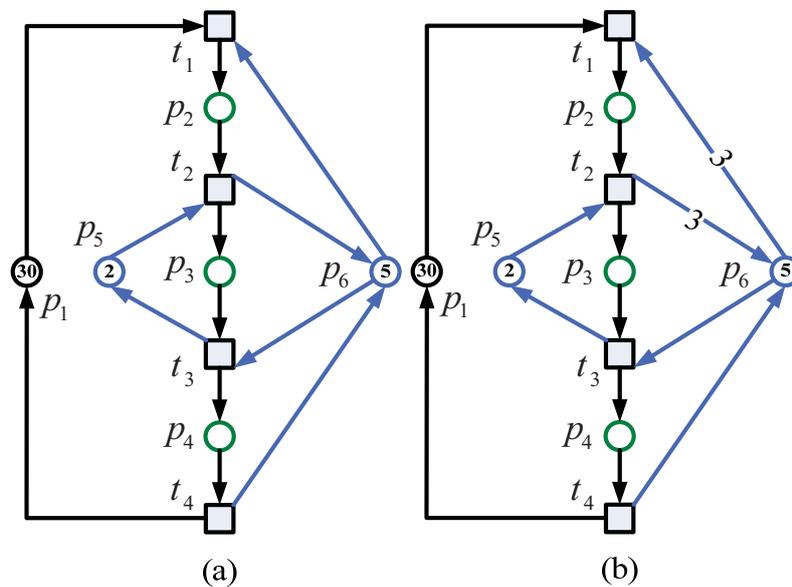


Fig. 9 (a) An S2PR (S3PR) net model and (b) a WS2PR (WS3PR) net model

Comme indiqué dans la Fig. 9, WS3PR est une extension des S3PR. Au lieu de se baser sur la notion centrale de siphon pour le contrôle du blocage, le travail ici est basé sur une autre structure remarquable notée ILS. Fig. 10 est un modèle S3PR model du problème des philosophes. La Fig. 11 donne une modélisation du même problème en termes de WS3PR. Comment assurer un contrôle simple et efficace dans le contexte des WS3PR est le point de départ de notre travail.

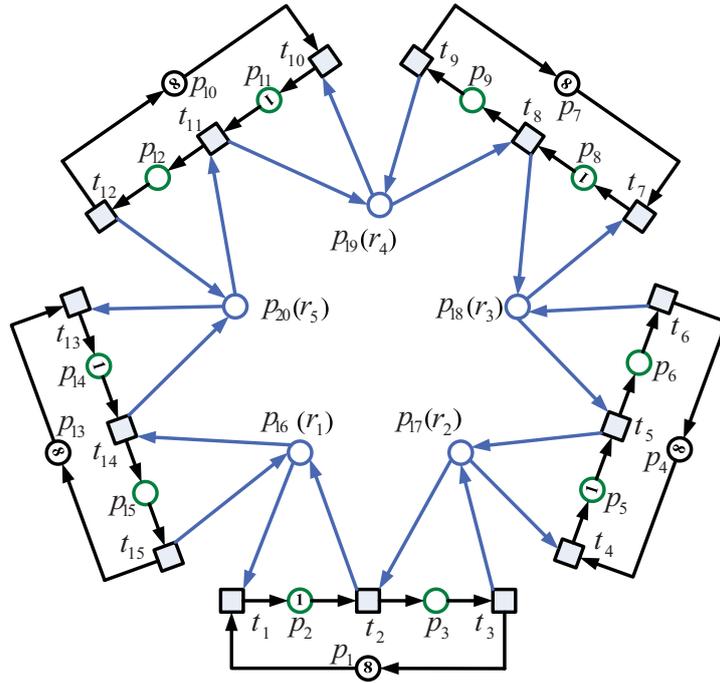


Fig. 10 An S3PR model for the problem of Dining Philosophers

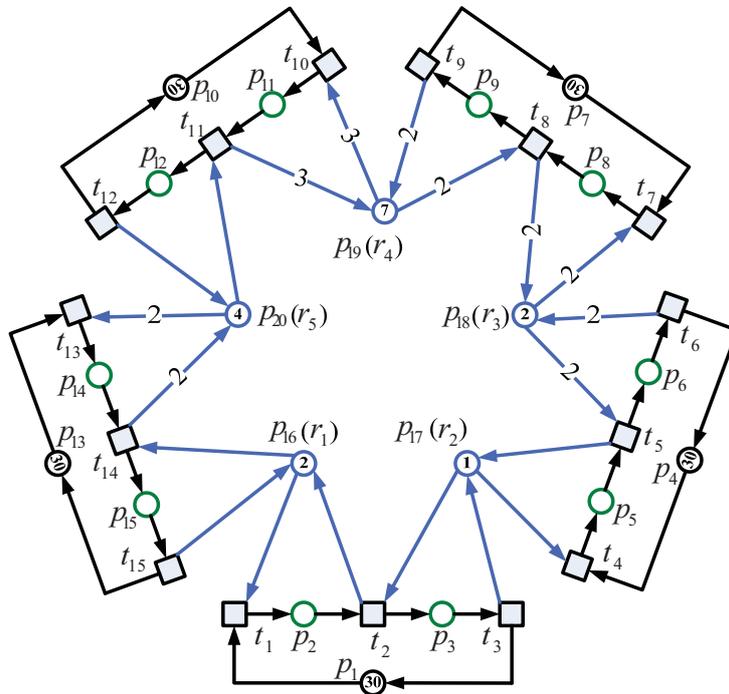


Fig. 11 A revised Dining Philosopher problem modeled by WS3PR with a unique WSDC

Le chapitre 4 présente une méthode originale et simple de contrôle d'un S3PR pour que celui-ci devienne vivant. Cette méthode est simple dans le sens où il suffit de contrôler

simplement l'état initial. Le point central est l'identification d'une condition suffisante de vivacité du modèle entre le marquage initial et les valuations des arcs en entrée et en sortie des places ressources. On montre que si cette condition est vérifiée, il n'existe pas de blocage circulaire dans une attente circulaire et ainsi, que le modèle est vivant. On détaille ensuite les algorithmes polynomiaux mettant en oeuvre la méthode et nous donnons quelques résultats expérimentaux. Ce résultat conduit à une méthode de contrôle peu contraignante. On termine ce chapitre par une discussion positionnant notre méthode vis-à-vis des autres méthodes, et démontrant ainsi son intérêt.

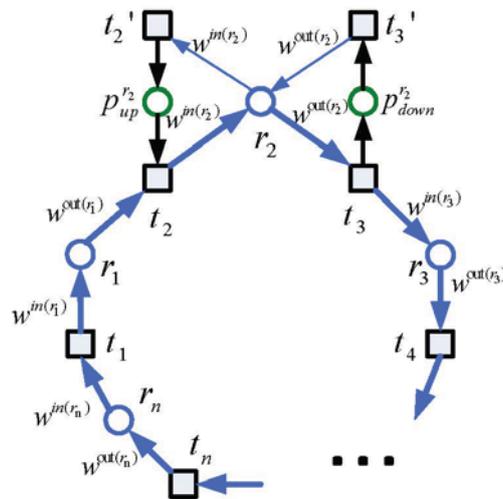


Fig. 12 A general WSDC.

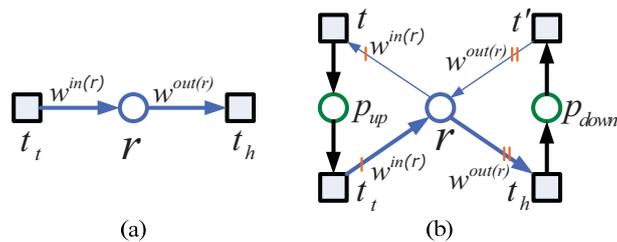


Fig. 13 (a) A competition path and (b) a subnet containing the competition path and its upstream and downstream activity places.

Cependant, il que le “competition path” est fréquent pour un système où plus de deux processus sont en compétition sur une ressource donnée (voir Fig. 14).

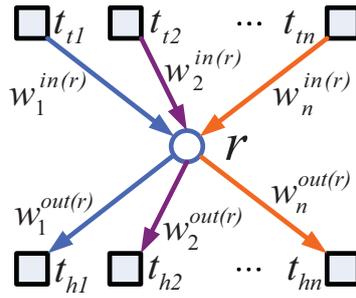


Fig. 14 Multiple competition paths with a same resource place.

Soient les deux WS3PRs décrits dans Figs. 15 and 16, qui présentent le meme probleme à savoir la forte competition sur une ressource par plusieurs processus. Ainsi, l'existence de nombreux « competition paths” qui nécessitent d’être contrôlés.

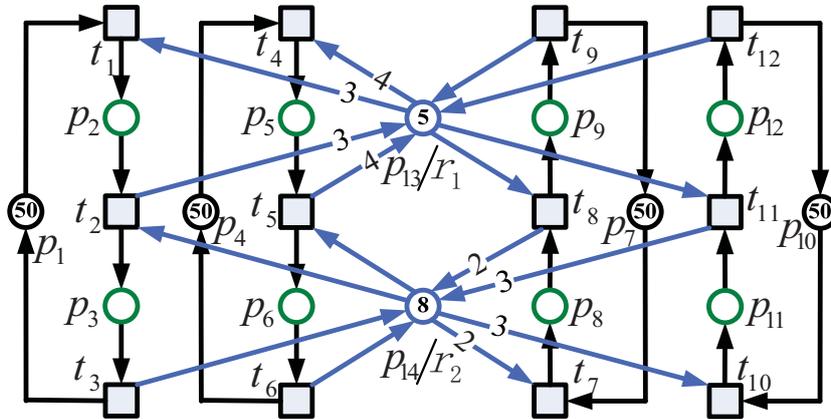


Fig. 15 A WS3PR net with two shared resource places

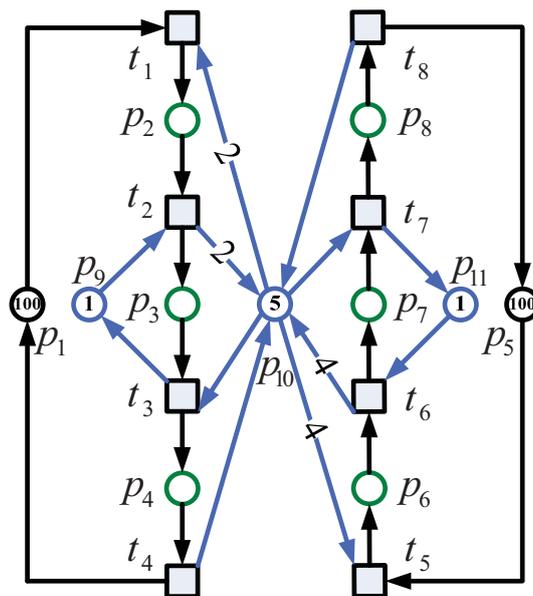


Fig. 16 A WS3PR net model and two competition paths with the resource place p10.

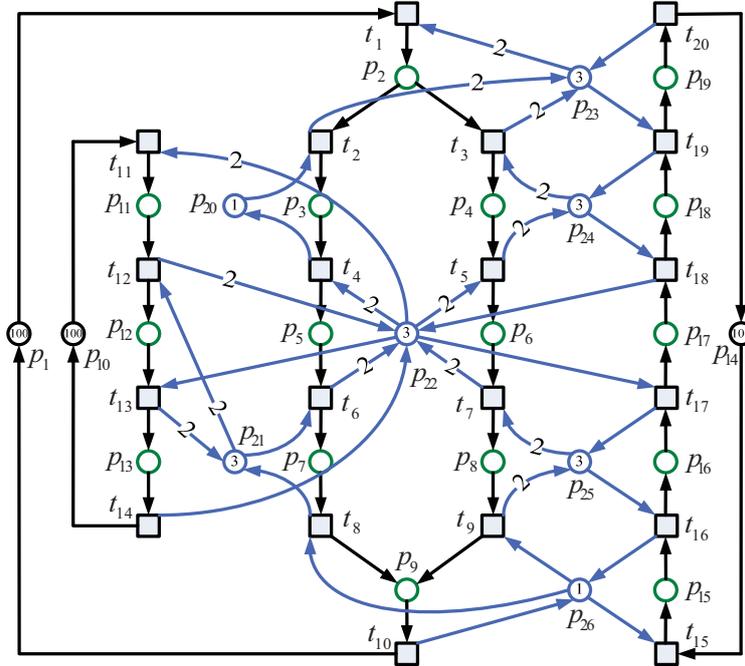


Fig. 17 A WS3PR model with 9 WSDC
all satisfying the proposed restriction
and the net is live

Dans le chapitre 5, On propose une méthode hybride combinant la méthode précédente et l'ajout de moniteur pour éviter les blocages circulaires. Le point clé de la méthode est la généralisation des méthodes classiques de contrôle des verrous applicables au S3PR.

On démontre qu'un WS3PR est vivant si tous les « circuits du graphe de ressources» (WSDC) contenu dans un verrou vérifient la contrainte exprimée dans le chapitre 4. On en déduit une méthode de contrôle de ces verrous garantissant la vivacité du modèle. Ce résultat étend la méthode précédente en offrant au concepteur diverses manières de contrôler un modèle pour rendre ce contrôle le plus permissif que possible.

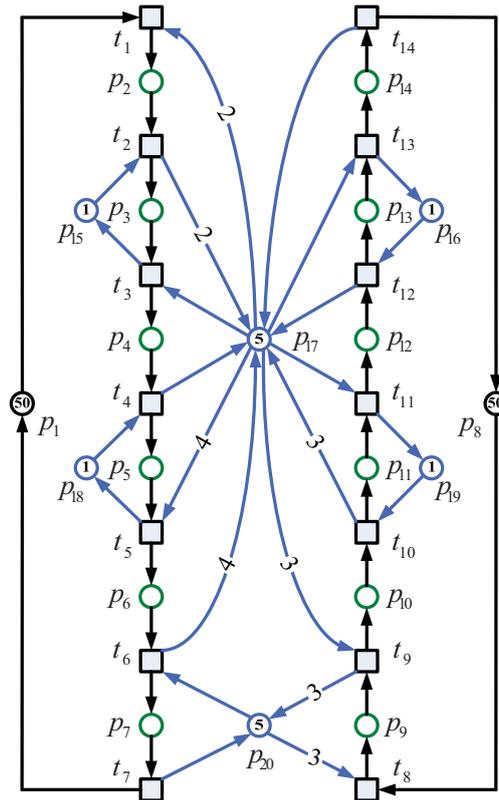


Fig. 20 A WS3PR net model with 6 resource places

Le chapitre 6 propose des techniques pour équilibrer l'usage des ressources entre les différentes activités du modèle. On introduit ainsi la notion de taux d'utilisation des ressources et on montre comment construire un superviseur pour que le modèle respecte ces nouvelles contraintes. Ce nouveau problème se ramène à la résolution d'un problème programmation linéaire en nombres entiers. Ce travail complète l'étude du chapitre précédent en répondant à des problèmes d'équilibrage de ressources que peuvent se poser des concepteurs de systèmes de production automatisée.

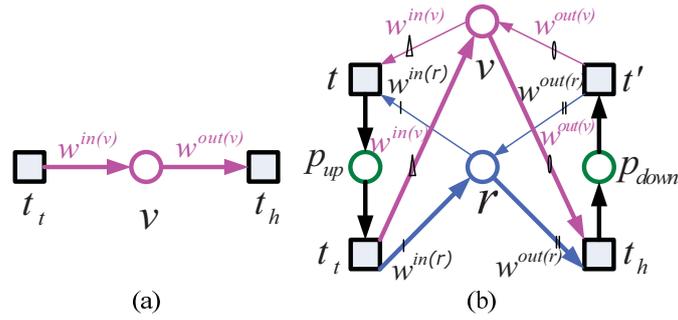


Fig. 21 (a) A control path and (b) the control path and corresponding paired arcs imposed on a subnet containing a competition path

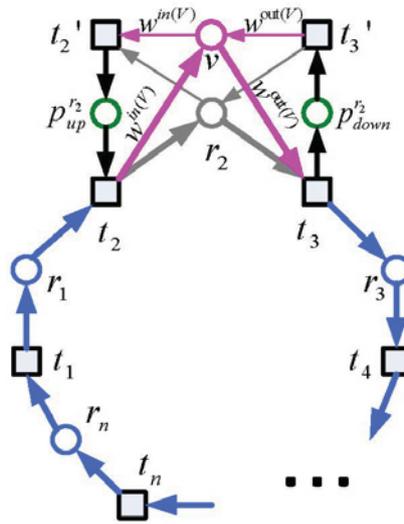


Fig. 22 A general WSDC and one of its LRS monitor

Le chapitre 7 s'attaque au problème du contrôle en décomposant le modèle en sous modèles. Le contrôle des éléments pathogènes, les circuits du graphe de ressources (WSDC), est réalisé sur chaque sous modèle. La justification de la méthode repose sur le fait qu'un circuit du graphe de ressources est localisé entièrement dans un sous modèle. On propose dans une discussion finale des méthodes plus fines de décomposition qui mériteraient d'être explorées et formalisées.

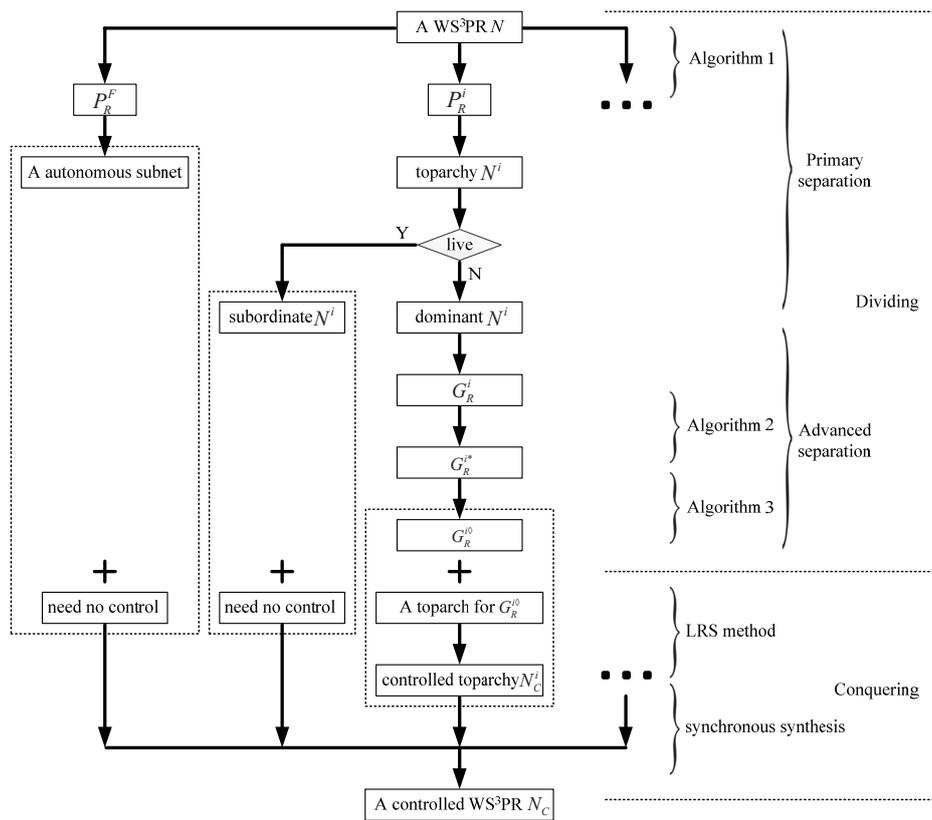


Fig. 23 A Schematic diagram of the D&C liveness-enforcing method for WS3PR

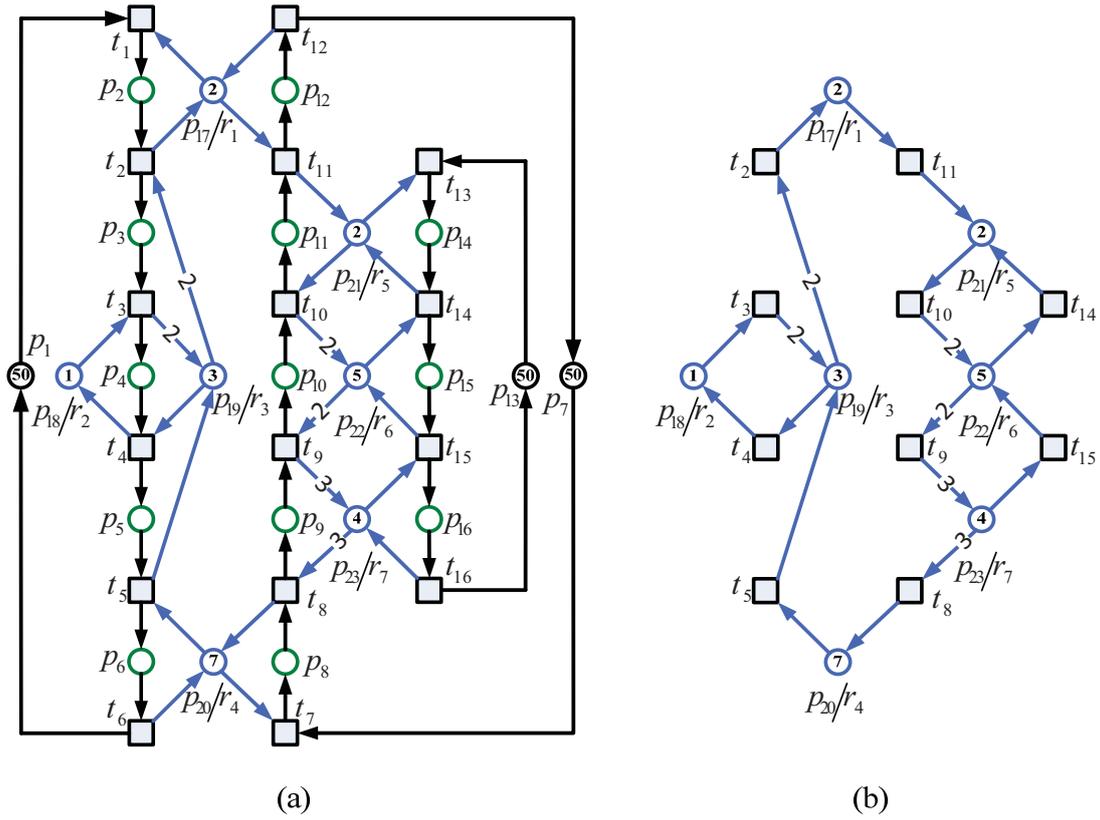


Fig. 24 (a) A WS3PR net and (b) its only one strongly connected graph

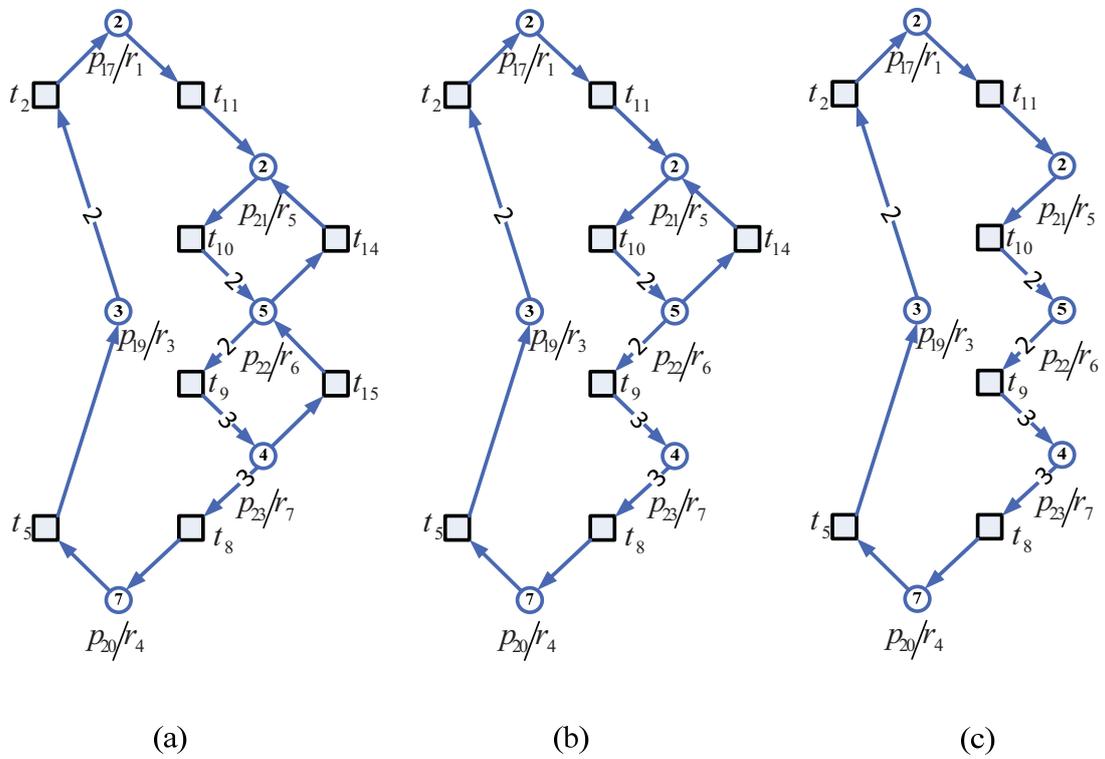


Fig. 25 (a), (b), and (c) are the gradually simplified subnet by using the proposed method

Le chapitre 8 est une application directe des résultats du chapitre 4. M. Ding LIU traduit la condition de vivacité des WS3PR sous la forme d'un programme linéaire en nombres entiers. Le résultat du programme induit un contrôle de l'allocation des ressources garantissant la vivacité du modèle. Ce chapitre contient quelques expérimentations sur un petit modèle démontrant l'intérêt de la méthode.

Afin d'éviter l'énumération de tous les WSDCs, On présente une méthode itérative basée sur une méthodes mixed integer programming (MIP).

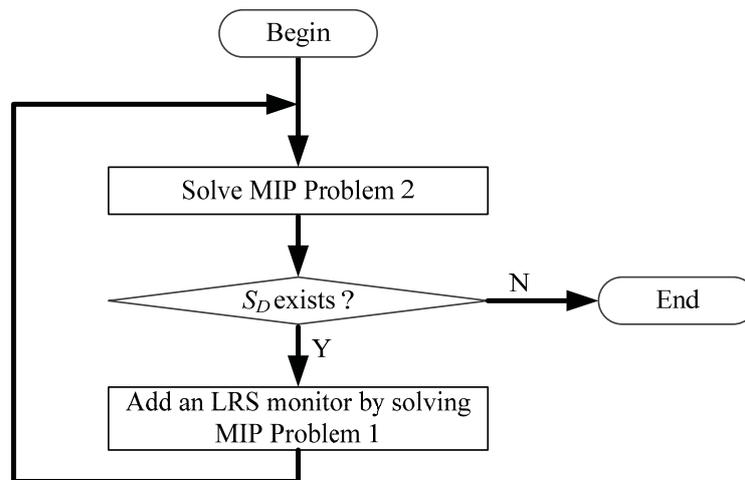


Fig. 26 A schematic diagram of the iterative method.

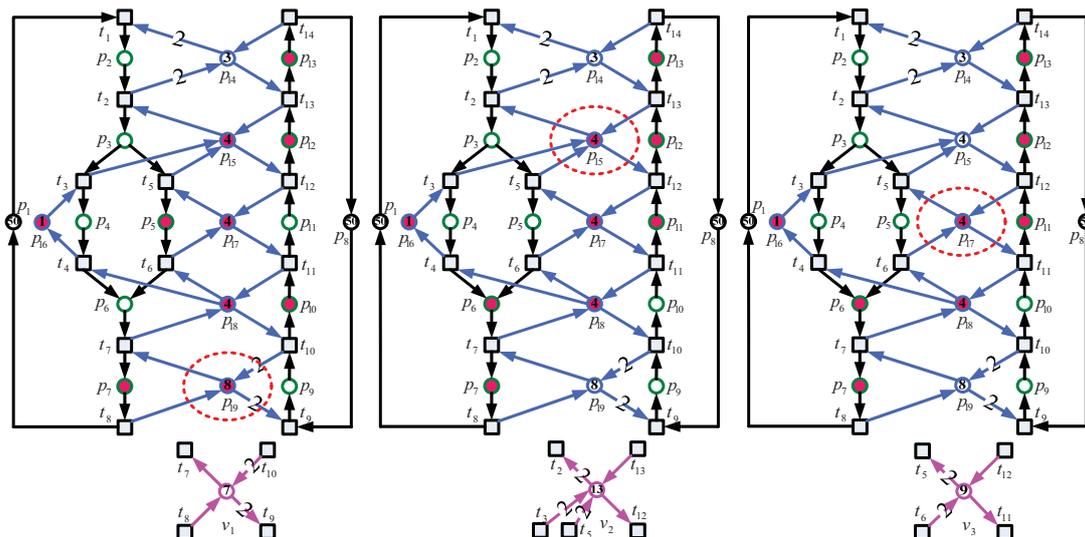


Fig. 27 An iterative control of a WS3PR net model with 3 LRS monitors

MIMS obtained by MIP	r_i	v_i	reachable states	dead states
/	/	/	3,334,653	30
$\{p_5, p_7, p_{10}, p_{12}, p_{13}, p_{15}, p_{16}, p_{17}, p_{18}, p_{19}\}$	p_{19}	v_1	2,663,888	6
$\{p_6, p_7, p_{11}, p_{12}, p_{13}, p_{15}, p_{16}, p_{17}, p_{18}\}$	p_{15}	v_2	2,613,824	1
$\{p_6, p_7, p_{11}, p_{12}, p_{13}, p_{16}, p_{17}, p_{18}\}$	p_{17}	v_3	2,500,037	0

Tab. 1 The control effect after each iteration

Le chapitre 9 propose une analyse de la vivacité sur une extension naturelle des WS3PR, les GS3PR. Dans les WS3PR, une activité peut utiliser plusieurs ressources d'un même type. Dans les GS3PR, une activité peut utiliser plusieurs ressources de différents types. M. Ding LIU donne une généralisation des résultats du chapitre 4 applicable au GS3PR. La formalisation du problème et de la solution sont bien posées. Les résultats semblent corrects et je regrette que le chapitre ne contienne pas de preuves.

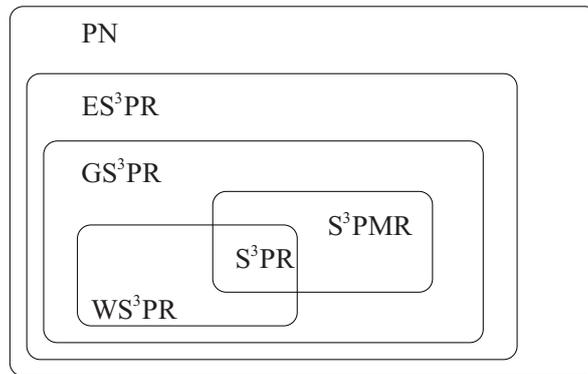


Fig. 28 Classes of Petri nets sorted according to process-oriented Petri nets and resource allocation systems

Tout le travail réalisé dans les chapitres précédents concerne la classe des WS3PR. On propose dans le chapitre suivant une extension de la méthode ILS aux GS3PR, une classe plus générale que celle WS3PR. Une classification of RdPs est donnée dans Fig. 28. Plus la classe est générale, plus le mode d'utilisation et de partage des ressources est complexe.

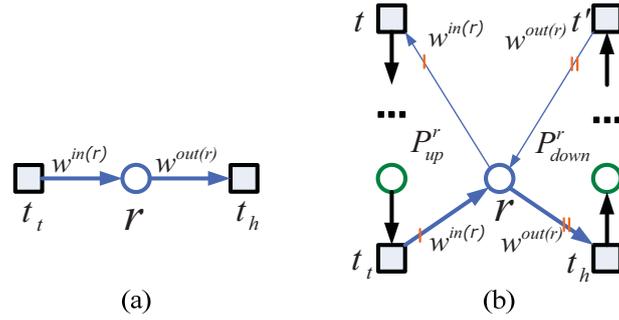


Fig. 29 (a) A competition path and (b) its general situation in GS3PR

Comme est décrit dans la Fig. 29(a), le “competition path” est le point de départ de notre analyse. Cependant, dans le cas de la Fig. 29(b), la structure à partir de laquelle un « competition path » peut être identifiée est plus complexe.

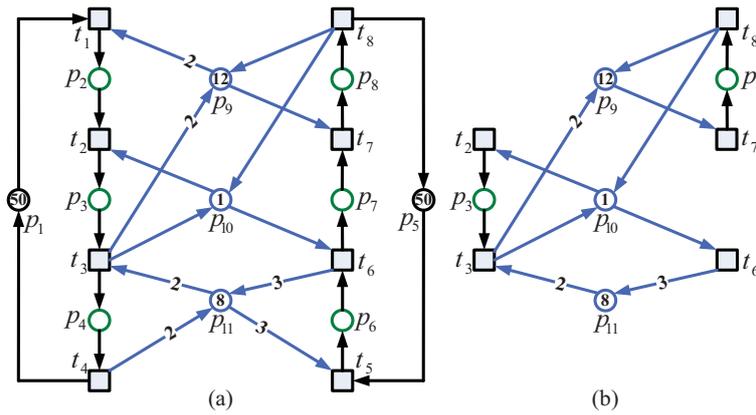


Fig. 30 (a) A marked GS3PR net model and (b) its weighted digraph of wait relations

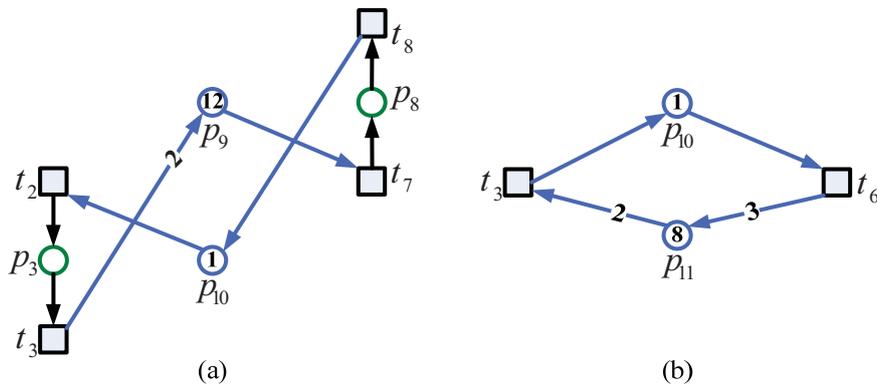


Fig. 31 Two WSDCs of the above GS3PR: (a) a WSDC including activity places and (b) another WSDC not including any activity place.

Résumé :

Nos travaux portent sur l'analyse des systèmes de production automatisée à l'aide de réseaux de Petri. Le problème posé est de savoir si un système peut se bloquer complètement ou partiellement et si besoin de calculer un contrôleur garantissant son bon fonctionnement. Les systèmes de production se modélisent naturellement à l'aide d'une sous-classe des réseaux de Petri, les S^3PR s. Ce modèle a été très largement étudié par le passé conduisant à des méthodes basées uniquement sur la structure du modèle. Dans ce travail, nous généralisons ces travaux aux modèles des WS^3PR , une extension des S^3PR où la réalisation d'une activité nécessite non par une ressource mais plusieurs ressources d'un même type et pour lesquels nous proposons des techniques originales combinant des éléments de théorie des graphes et de théorie des nombres, améliorant même les méthodes du passé sur le modèle simple des S^3PR .

On présente une caractérisation fine de la vivacité d'un tel modèle basée la notion d'attente circulaire. Une attente circulaire peut être vue comme une composante connexe du sous graphe réduit aux transitions et aux places ressources du modèle. Puis nous démontrons que la non vivacité d'un WS^3PR est équivalente à l'existence d' « un blocage circulaire dans une attente circulaire ». Ce résultat généralise finement la caractérisation de la vivacité d'un S^3PR . Après avoir introduit la notion de « circuits du graphe de ressources » (WSDC), on construit une méthode de contrôle de ces verrous garantissant la vivacité du modèle d'autant plus efficace qu'une méthode de décomposition du réseau est proposée. Enfin, une traduction de traduit la condition de vivacité des WS^3PR sous la forme d'un programme linéaire en nombres entiers est établie et des expérimentations ont démontré l'intérêt de la méthode pour contrôle de systèmes l'allocation des ressources.

Mots clés :

Systèmes de production flexible, Réseaux de Petri généralisés, Analyse structurelle, Vivacité structurelle, Contrôle d'interblocage

Abstract :

As an indispensable component of contemporary advanced manufacturing systems, flexible manufacturing systems (FMSs) possess flexibility and agility that traditional mass production systems lack. An FMS usually consists of picking and placing robots, machining centers, logistics systems, and advanced control systems. Some of them can be recognized as the FMS' shared resources, which result in its flexibility but may also lead to its deadlocks. As a classic problem in resource allocation systems, deadlocks may arise in a fully automated FMS and bring about a series of disturbing issues, from degraded and deteriorated system productivity and performance to low utilization of some critical and expensive resources and even long system downtime. Therefore, the analysis of and solution to deadlock problems are imperative for both a theoretical investigation and practical application of FMSs. Deadlock-freedom means that concurrent production processes in an FMS will never stagnate. Furthermore, liveness, another significant behavioral property, means that every production process can always be finished in finite time. Liveness implies deadlock-freedom, but not vice versa. The liveness-enforcement is a higher requirement than deadlock-freedom.

Completely without reachability graph, the work is based on the logical reasoning of model structures. From the perspective of the behavioral logic, the thesis focuses on the intrinsically live structures and deadlock control of generalized Petri nets modeling flexible manufacturing systems. Being different from the prevailing structure/siphon-based methods, a concept of intrinsically live structures becomes the starting point to design, analyze, and optimize a series of novel deadlock control and liveness-enforcing methods in the work. The characteristics and essence of intrinsically live structures are identified and derived from subclasses of generalized Petri nets modeling FMSs with complex resource usage styles. In addition, the numerical relationship between initial markings and weights of connecting arcs is investigated and used to design restrictions that can ensure the intrinsic liveness of global or local structures. With the structural theory, graph theory, and number theory, the work in the thesis achieves the goals of deadlock control and liveness-enforcement. The proposed methods are superior over the traditional siphon-based ones with a lower computational complexity (or a higher computational efficiency), a lower structural complexity, and a better behavioral permissiveness of the controlled system.

Keywords :

flexible manufacturing system, Petri net, generalized Petri net, live structure, deadlock control