Supervisor Synthesis for Automated Manufacturing Systems Based on Structure Theory of Petri Nets

Gaiyun Liu

CONSERVATOIRE NATIONAL DES ARTS ET MÉTIERS

le cnam

École Doctorale Informatique, Télécommunications et Electronique de Paris

Centre d'Etudes et De Recherche en Informatique du CNAM

THÈSE DE DOCTORAT

présentée par : **Gaiyun LIU**

soutenue le : **27 Novembre 2014**

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

Discipline / Spécialité : **Informatique**

Supervisor Synthesis for Automated Manufacturing Systems Based on Structure Theory of Petri Nets

THÈSE DIRIGÉE PAR

M. BARKAOUI Kamel	*Professeur, CEDRIC, Cnam, Paris*
M. LI Zhiwu	*Professeur, SCAG, Xidian Université, Chine*

RAPPORTEURS

Mme. CHU Feng	*Professeur, IBISC, Université d'Evry Val d'Essonne*
M. HANISCH Hans-Michael	*Professeur, Martin Luther Universität of Halle–Wittenburg*

EXAMINATEURS

Mme. BÉRARD Béatrice	*Professeur, LIP6, Université Pierre et Marie Curie*
M. BOIMOND Jean Louis	*Professeur, LARIS, Université Nantes Angers Le Mans*
M. RAÏSSI Tarek	*Professeur, CEDRIC, Cnam, Paris*

Remerciements

Foremost, I would like to express sincere gratitude to my advisor, Prof. Kamel Barkaoui, for his constant encouragements and guidance. My sincere thanks go to him since the publications of my papers are impossible without his valuable suggestions, critical comments and reviews, as well as sweet encouragements and kind support.

Second, I would like to express my heartfelt gratitude to my co-advisor, Prof. Zhiwu Li, Xidian University. He has helped me in a variety of ways since 2008. His important and useful directions greatly improve the development of my original idea on supervisory control in DES. He patiently corrected my writing and financially supported my research. He has walked me through all the stages of the writing of this dissertation.

Besides my advisors, I would like to thank the rest of my thesis committee : Prof. Béatrice Bérard (examiner), Prof. Jean Louis Boimond (examiner), Prof. Feng Chu (reviewer), Prof. Hans-Michael Hanisch (reviewer), and Prof. Tarek Raïssi (examiner) for their time, patience, encouragement, insightful comments, and interesting questions.

I extend very special thanks to many people who directly or indirectly contribute in a variety of ways to the development of the material included in this dissertation. The continuing interaction and stimulating discussions with them have been a constant source of encouragement and inspiration. They include Professors Daniel Yuh Chao, Taiwan Cheng Chi University, Yisheng Huang, Taiwan ILan University, Murat Uzam, Meliksah Universitesi, Long Wang, Peking University, Naiqi Wu, Guangdong Institute of Technology, Keyi Xing, Xi'an Jiaotong University, and Mengchu Zhou, New Jersey Institute of Technology.

Moreover, I would like to thank all the members of VESPA at Le CNAM for the immeasurable assistance they have given me over the years. I am truly grateful to all the fellows in the laboratory at Xidian University. It has been a pleasure to conduct research with them in System Control & Automation Group.

Finally, my thanks would go to my beloved family for their loving considerations and great confidence in me all through these years. I also owe my sincere gratitude to my friends who gave me their help and time in listening to me and helping me work out my problems during my life in Paris.

Résumé

Le contrôle de systèmes industriels à cause de l'automatisation et la réduction de nombre des opérateurs devient un enjeu crucial. Les systèmes de production automatisés (AMS) sont d'autant plus touchés car une défaillance du programme de contrôle peut réduire considérablement la productivié voire entraôner l'arrêt du système de production. Pour certains de ces systèmes où le partage des ressources est pondérant, la notion de blocage partiel ou global est fréquente et la validation avant implantation est préférable pour réduire les risques.

En raison de la capacité des réseaux de Petri à décrire aisément l'exécution concurrente des processus et le partage des ressources, de nombreuses méthodes de vérification d'absence de blocage et de synthèse de contrôleurs basées sur la théorie structurelle ou le graphe d'accessibilité des réseaux de Petri ont été proposées au cours des deux dernières décennies. Traditionnellement, une méthode de prévention de blocage est évaluée selon trois critères de performance : la complexité structurelle, la permissivité comportementale, et la complexité de calcul. Les méthodes fondées sur l'espace d'état aboutissent généralement à un contrôle maximal permissif mais souffrent de l'explosion combinatoire de l'espace d'états. En revanche, les méthodes de synthèse de contrôleurs fondées sur l'analyse structurelle évitent le problème de l'explosion de l'espace d'état mais aboutissent à des superviseurs pouvant restreindre considérablement les comportements admissibles du système. De plus si la théorie structurelle de contrôle de siphons pour la synthèse des superviseurs est mature dans le cas des réseaux de Petri ordinaires, elle est en développement pour les réseaux de Petri généralises. Par ailleurs, la plupart des travaux existants partent du principe que les ressources sont constamment disponibles. Or l'indisponibilité de ressources est en réalité un phénomène ordinaire. Il serait donc judicieux de développer une politique de vérification de blocage qui soit efficace tout en considérant des ressources non fiables.

Cette thèse vise principalement à faire face aux limitations mentionnées ci-dessus. Nos princi-

pales contributions à la fois théoriques et algorithmiques sont les suivantes.

Premièrement, après avoir revisité les conditions de contrôlabilité des siphons (cs-propriété) et précisé les limitations des max, max′ et max′′-cs- propriétés, nous définissons la max*-cs-propriété et nous démontrons que cette nouvelle propriété est une condition non seulement suffisante mais aussi nécessaire pour la vivacité de la classe des GS^{3}PR (Generalized Systems of Simple Sequential Processes with Resources). Par la suite nous montrons comment le problème de la vérification de cette propriété et donc la vivacité des GS^{3}PR peut se ramener à la résolution d'un programme linéaire en nombre entiers.

Dans une seconde partie, nous proposons une classe de réseaux de Petri appelée M-Nets dotée d'une forte capacité de modélisation des systèmes de production automatisés. En combinant la théorie du contrôle siphon avec la théorie des régions, nous développons une méthode de prévention de blocage ayant un bon compromis entre l'optimalité du comportement et la complexité de calcul. De plus, nous proposons une méthode de synthèse d'un contrôleur maximal permissif pour une sous-classe de réseaux notée β-nets basée sur des distributions de jetons dans les siphons et évitant la génération du graphe d'accessibilité ainsi que l'énumération des siphons minimaux.

Enfin, nous proposons dans cette thèse une méthode de conception d'un superviseur de systèmes de production automatisés où les ressources ne sont pas toutes fiables et particulièrement efficace pour les systèmes pouvant être modélisés par les réseaux S^{3}PR (Systems of Simple Sequential Processes with Resources).

Mots clés : Systèmes de production automatisés, réseaux de Petri, blocage, siphon, synthèse de contrôleurs.

Abstract

Because of automation and reduction of the number of operators, the control of industrial systems is becoming a critical issue. For automated manufacturing systems (AMS) where resource sharing is preponderant, the notion of partial or total blocking is frequent and validation before implementation is preferable to reduce the risks.

Due to the easy and concise description of the concurrent execution of processes and the resource sharing by Petri nets, many methods to verify deadlock-freeness and to synthesize controllers using structural theory or reachability graph have been proposed over the past two decades. Traditionally, a deadlock control policy can be evaluated by three performance criteria : structural complexity, behavioral permissiveness, and computational complexity. Generally, deadlock control policies based on the state space analysis can approach the maximal permissive behavior, but suffer from the state explosion problem. On the contrary deadlock control policies based on the structural analysis of Petri nets avoid in general the state explosion problem successfully, but cannot lead to the maximally or near maximally permissive controller. Moreover, the current deadlock control theory based on siphons is fairly mature for ordinary Petri nets, while for generalized Petri nets, it is presently at an early stage. On the other hand, most deadlock control policies based on Petri nets for AMS proceed on the premise that the resources in a system under consideration are reliable. Actually, resource failures are inevitable and common in most AMS, which may also cause processes to halt. Therefore, it is judicious to develop an effective and robust deadlock control policy considering unreliable resources.

This thesis aims to cope with the limitations mentioned above. Our main theoretical and algorithmic contributions are introduced as the following.

Firstly, after revisiting the controllability conditions of siphons and limitations of max, max′,

and max″-controlled siphon properties (cs-properties), we define the max*-cs-property and prove that this new cs-property is not only sufficient but also a necessary liveness condition for generalized systems of simple sequential processes with resources (GS^3PR). Moreover, we show the verification of this property and hence liveness of GS^3PR nets can be translated into resolution of an integer programming (IP) model.

Secondly, we propose a class of manufacturing-oriented Petri nets, M-nets for short, with strong modeling capability. Combining siphon control and the theory of regions, we develop a deadlock prevention method that makes a good trade-off between behavioral optimality and computational tractability. Moreover, this thesis proposes a maximally permissive control policy for a subclass of Petri nets (called β-nets) based on the token distribution patterns of siphons and avoiding the generation of reachability graphs and enumeration of minimal siphons.

Finally, we propose a design method of robust liveness-enforcing supervisors for AMS with unreliable resources. The proposed method is appropriate in particular for plants which can be modeled by systems of simple sequential processes with resources (S^3PR).

Keywords : Automated manufacturing system, Petri net, deadlock, siphon, controller synthesis.

Table des matières

Remerciements **3**

Résumé **5**

Abstract **7**

Liste des tableaux **13**

Liste des figures **17**

Introduction **19**

1. Background 19

2. Literature Review 22

2.1 Liveness-enforcing Supervision for Automated Manufacturing Systems . . . 22

2.2 Automated Manufacturing Systems with Unreliable Resources 29

3. Thesis Organization 30

1 Preliminaries of Petri Nets **33**

1.1 Introduction 33

1.2 Formal Definitions 33

1.3 Deadlocks and Livelocks 38

1.4 Inhibitor Arc 40

1.5 Structural Invariants, Siphons, and Traps 41
1.6 Subclasses of Petri Nets 43
1.6.1 S^3PR Net 45
1.6.2 GS^3PR Net 49
1.6.3 S^4R Net 53
1.6.4 Relationships among S^3PR, GS^3PR, and S^4R Nets 54
1.7 Summary 55

2 Structural Analysis of GS^3PR Nets 57

2.1 Introduction 57
2.2 Motivation 58
2.2.1 Max-controlled Siphons 58
2.2.2 Max $'$-controlled Siphons 59
2.2.3 Max$''$-controlled Siphons 60
2.3 Necessary and Sufficient Condition 62
2.4 Comparison 69
2.5 Liveness Detection for GS^3PR 70
2.6 Examples and Discussions 74
2.7 Summary 81

3 Deadlock Prevention for M-nets Based on Structure Reuse of Supervisors 83

3.1 Introduction 83
3.2 Structure Design of a Petri Net Supervisor 85
3.2.1 Motivation and Problem Formulation 85
3.2.2 M-nets 87
3.2.3 Minimal Initial Marking 89
3.2.4 Derivation of the Structure of a Controlled System 91

3.3 Siphon Controllability Constraints 92
3.4 Redundancy Identification of Constraints 98
3.5 Deadlock Prevention Policy 100
3.6 Examples 102
3.7 Discussions 105
3.8 Summary 106

4 Maximally Permissive Control Policy for a Subclass of S^3PR without Reachability Analysis 109

4.1 Introduction 109
4.2 Critical Siphon 110
4.3 Control Policy 116
4.4 Examples 119
4.5 Summary 126

5 Robustness of Deadlock Control for S^3PR with Unreliable Resources 129

5.1 Introduction 129
5.2 Motivation 130
5.3 Robust Liveness-enforcing Supervisor Design 132
5.3.1 Liveness-enforcing Supervisor Design 133
5.3.2 Robustness of a Supervisor 140
5.3.3 Improvement of Algorithm 5.2 148
5.4 Examples 151
5.5 Discussions 154
5.6 Summary 157

Conclusions and Future Research 159

1. Contributions 159

2. Limitations and Future Research 161

Bibliographie **163**

Glossaire **179**

Publications **181**

Liste des tableaux

2.1 Controllability of siphon $\{p_2, p_4, p_6, p_8, p_9, p_{10}\}$ at different markings 62

2.2 Controllability conditions of siphon $\{p_2, p_4, p_6, p_8, p_9, p_{10}\}$ in Fig. 2.1(a) 70

3.1 Redundancy conditions of the dependent constraints in Fig. 3.2(b) 99

3.2 Behavioral permissiveness of the proposed deadlock prevention policy in Fig. 3.4(b) 103

3.3 Behavioral permissiveness of the proposed deadlock prevention policy in Fig. 3.6(b) 104

4.1 Basic siphons, resource circuits, and $V(M_0)$ for the net in Fig. 4.1(a) 111

4.2 Controlled model for the net in Fig. 4.1. 117

4.3 Controlled model for the net in Fig. 4.2. 121

4.4 Types of siphons, their dependency on basic siphons, $[S]$, UP, and EUP for the net in Fig. 4.3 122

4.5 Controlled model for the net in Fig. 4.3. 123

4.6 Basic siphons, resource circuits, and $V(M_0)$ for the net in Fig. 4.4 124

4.7 Compound siphons, their dependency on basic siphons, UP, and EUP for the net in Fig. 4.4 124

4.8 Controlled model for the net in Fig. 4.4. 125

5.1 Monitors in the toparches for three toparchies 154

Table des figures

1.1 A Petri net (N, M_0). 35

1.2 The reachability graph of net (N, M_0) shown in Fig. 1.1. 37

1.3 A Petri net (N, M_0) in [81]. 39

1.4 An example of livelocks shown in a reachability graph. 39

1.5 An extended Petri net with an inhibitor arc. 40

1.6 (a) A manufacturing system, (b) its net model (N, M_0) 44

1.7 (a) An S^3PR, (b) a GS^3PR, (c) an S^4R. 45

1.8 (a) An S^2P, (b) a marked Petri net (N_1, M_{10}), (c) a marked Petri net (N_2, M_{20}), (d) the composed S^3PR. 46

1.9 (a) A marked Petri net (N_1, M_{10}), (b) a marked Petri net (N_2, M_{20}), (c) the composed GS^3PR. 49

1.10 Typical structure of a siphon in GS^3PR (N, M_0). 52

1.11 An S^4R net (N, M_0). 54

2.1 (a) A GS^3PR net (N, M_0), (b) a live GS^3PR with a non-max-controlled siphon. . . 59

2.2 A live GS^3PR with a non-max′-controlled siphon. 60

2.3 A live GS^3PR with a non-max ″-controlled siphon. 62

2.4 (a) A GS^3PR (N, M_0), (b) the net at $M = 6p_1 + p_2 + 2p_3 + p_7 + p_8 + 9p_{10} + p_{12} + p_{13} + p_{17} + 9p_{19}$. 66

2.5 A net in [81]. 78

2.6 The reachability graph of the net in Fig. 2.5. 78

3.1 (a) A plant model (N, M_0), (b) the reachability graph of (N, M_0), (c) a modified model (N^m, M_0^m), (d) the reachability graph of (N^m, M_0^m), (e) a controlled system (N^{mc}, M_0^{mc}) for (N^m, M_0^m), (f) a controlled system (N^c, M_0^c) for (N, M_0). 86

3.2 (a) A plant model (N, M_0), (b) a controlled system (N^{mc}, M_0^{mc}) for (N, M_0). 93

3.3 Flowchart of the deadlock prevention policy. 101

3.4 (a) An M-net (N, M_0), (b) controlled system (N^{mc}, M_0^{mc}). 102

3.5 (a) Layout of an AMS, (b) routes of part types P1 and P2. 103

3.6 (a) Petri net model of an AMS, (b) structure of the controlled system. 104

3.7 (a) An example net, (b) its controlled system. 105

4.1 An example of the control policy based on UP but with no unmarked places in $H(V)$. 111

4.2 Another example of the control policy based on UP but with no unmarked places in $H(V)$. 119

4.3 A more complicated Petri net model of an AMS. 121

4.4 An S^3PR model in [23]. 123

5.1 (a) An automated manufacturing cell, (b) a Petri net model, (c) a liveness-enforcing supervisor. 130

5.2 (a) One robot is removed, (b) one token is correspondingly removed from p_6. . . 131

5.3 A robust supervisor for a system. 132

5.4 An S^3PR (N, M_0). 134

5.5 (a) An autonomous subnet, (b) a toparchy of the net in Fig. 5.4 (a). 137

5.6 The idle subnet (N^{id}, M_0^{id}). 138

5.7 Monarch synthesis for (N, M_0) in Fig. 5.4 (a). 140

5.8 A recovery subnet. 142

5.9 Transformation. 148

5.10 Supervisor with inhibitor arcs. 150

5.11 (a) An AMS's layout, (b) the production routings of the AMS. 151

5.12 A robust supervisor for the original system. 152

5.13 Plant model (N, M_0). 153

5.14 A robust supervisor for the original system. 155

Introduction

Technological revolution in our real-world increasingly requires new techniques for the synthesis and verification of complex systems such as automated manufacturing systems (AMS), communication networks, communication systems, and traffic control systems. An AMS is readily represented in a logical form as discrete event systems (DES). Petri nets are well suitable to describe AMS' behavior and characteristics such as concurrency, conflict, and causal dependency. They can be used to reveal such behavioral properties as liveness, and boundedness via a Petri net formalism [16], [24]. Compared to finite state automata that are extensively used in the DES framework, Petri nets offer a compact representation of DES, as they do not represent explicitly the state space of the system.

This thesis addresses new methodologies for the supervisory control of AMS. This introductory chapter first provides the background of AMS and their deadlock problems and resolution. Subsequently, liveness-enforcing supervision for AMS with and without unreliable resources are briefly discussed and research issues to be addressed are exposited. Finally, the thesis organization is presented.

1. Background

Competition among the world's major industrial nations has renewed interest in the issues of increasing productivity through state-of-the-art manufacturing technologies. Such a technological edge can be achieved through the development and deployment of advanced automated manufacturing systems (AMS). An AMS is a new type of manufacturing pattern with a computer-controlled configuration to automatically produce different products. There are three main systems in most AMS : (1) work machines to perform a series of operations, (2) an integrated material transport

system with a computer to control the flow of materials, tools, and information throughout the system, and (3) auxiliary work stations for loading and unloading, cleaning, inspection, etc.

To effectively utilize the precious resources, they must be shared and carefully coordinated among various competing jobs. The high level of resource sharing may lead to circular wait conditions, the cause of deadlocks in which each of a set of two or more jobs keeps waiting indefinitely for the other jobs in the set to relinquish resources that they hold. Deadlocks and related blocking phenomena can give rise to unnecessary productivity loss, and even catastrophic results in some highly automated systems such as semiconductor manufacturing. It is therefore necessary to explore an effective and computationally efficient mechanism to properly allocate resources such that deadlocks can never occur. With the wide application of AMS, their deadlock control problem has been extensively studied over the last two decades, leading to significant theoretical results and successful industrial applications [16], [23], [46], [48], [71], [69], [84], [104].

Generally speaking, deadlocks in an AMS are considered to be a result of (1) shortage of system resource, (2) improper order of process execution, and (3) misallocation of resources. In summary, there are four conditions for a deadlock to occur, known as the Coffman conditions [17].

1. Mutual exclusion condition : a resource that cannot be used by more than one process at a time ;
2. Hold and wait condition : processes already holding resources may request new resources held by other processes ;
3. No pre-emption condition : no resource can be forcibly removed from a process holding it, resources can be released only by the explicit action of the process ;
4. Circular wait condition : two or more processes form a circular chain where each process waits for a resource that the next process in the chain holds.

The first three conditions are necessary but not sufficient for a deadlock to exist. For a deadlock to actually take place, the fourth condition is required. That is to say, once a deadlock occurs, all the four conditions must hold. On the contrary, a deadlock will never occur if one of these conditions is not satisfied.

Removing the “mutual exclusion" condition means that no process may have exclusive access to a resource. This proves impossible for resources that cannot be spooled, and even with spooled

resources deadlock could still occur.

The “hold and wait" condition may be removed by requiring processes to request all the resources they will need before starting up. This advance knowledge is frequently difficult to satisfy and, in any case, is an inefficient use of resources. Another way is to require processes to release all their resources before requesting all the resources they will need. This is often impractical.

The “no pre-emption" condition may also be difficult or impossible to avoid as a process has to be able to have a resource for a certain amount of time, or the processing outcome may be inconsistent or thrashing may occur. However, the inability to enforce pre-emption may interfere with a priority algorithm.

The “circular wait" condition may be prevented by establishing off-line a precedence to each resource and forcing processes to request resources in order of increasing precedence. This forces resource allocation to follow a particular and non-circular ordering. Hence, circular wait cannot occur.

The necessity of the four conditions for a deadlock to occur leads us to infer that negating one of them makes impossible the occurrence of deadlocks in an AMS. The physical characteristics and technical background of an AMS show that the first three deadlock conditions always hold and the only feasible doorway to eliminate deadlocks is to falsify the circular wait condition [35].

Deadlocks can be addressed by different approaches, generally classified into three categories : deadlock detection and recovery [109], [119], [120], [130], deadlock prevention [3], [23], [46], [35], [69], [56], and deadlock avoidance [109], [2], [94], [113], [114], [89], [115], [116], [117].

The deadlock detection and recovery is an optimistic strategy that grants a resource to a request as long as it is available. A detection algorithm is used to detect the occurrence of deadlocks. Once detected, a recovery mechanism is initialized by aborting one or more processes involved in a deadlock and the resources held by the aborted processes are relinquished. A deadlock detection and recovery strategy is often used in the case where deadlocks are infrequent and their consequence is not serious. It should be noted that this strategy is in general undesirable based on the performance studies [131].

In the deadlock avoidance strategy, a resource is granted to a process only if the resulting state is not a deadlock. In order to decide whether the forthcoming state is safe after a resource is

allocated to a process, every cell controller and global controller need to keep track of the global system state. Some aggressive deadlock avoidance policies do not eliminate all deadlock states [113].

The deadlock prevention is a static strategy that imposes restrictions on the interactions among resources and processes such that resource requests that may lead to deadlocks are prevented. Deadlock prevention does not suffer from the danger of system stoppage and thus is used extensively.

The synthesis and implementation of a deadlock resolution policy can be done based on a number of different formal models of AMS such as digraphs, automata, and Petri nets [35]. Digraphs are a simple and intuitive tool to describe interactions between operations and resources, from which a deadlock control policy can be derived. The representative research groups are led by Wysk [119], [118], and Fanti [26], [27], [28], [29], [30], [31], [32], [33], [34]. Based on formal languages and finite automata, supervisory control theory (SCT) [92] originated by Ramadge and Wonham provides a comprehensive and structural treatment of the modeling and control of DES. As an important paradigm, SCT has a profound influence on the supervisory control of AMS under other formalisms such as Petri nets. A number of effective yet computationally efficient deadlock control policies are developed based on automata. Lawley, Reveliotis, and Ferreira are distinguished experts in this area [59], [60], [61], [62], [63], [64], [65], [66], [67], [93].

Petri nets have been widely used for modeling, analysis, and deadlock control of AMS. They are well suitable to describe AMS' behavior and characteristics such as concurrency, conflict, and causal dependency. They can be used to reveal such behavioral properties as liveness, and boundedness [6], [16], [24]. Based on Petri nets, various deadlock resolution strategies are developed.

2. Literature Review

2.1 Liveness-enforcing Supervision for Automated Manufacturing Systems

In recent years, liveness-enforcing supervisory control has been an active area of research for AMS characterized by processes with highly ordered, linear workflows. Petri nets can describe AMS in a quite compact way. Over the past two decades, many researchers considered Petri nets as an alternative to automata. Deadlock control policies in the framework of Petri nets can be

developed on the basis of state space analysis, structural analysis, and combination of the former two methods. Traditionally, a deadlock control policy can be evaluated by a number of performance criteria : structural complexity, behavioral permissiveness, and computational complexity.

State space analysis considers the reachable space of a net model that reflects possible behavior of system evolution. The studies in [1], [104], [38], [105], [106], [107], [108], and [14] are some of the most representative works.

In [1], the theory of regions is proposed. It originally aims to provide a formal methodology to synthesize a Petri net from a transition system. Later, Uzam [104] proposes an approach to design optimal Petri net supervisors by using the theory of regions. Shortly after the study in [104], Ghaffari *et al.* present an easily understandable explanation of the design approach to an optimal liveness-enforcing Petri net supervisor based on the theory of regions [38] by using linear algebra. The work in [105] is an improved version of the study in [104].

If an optimal supervisor exists for a Petri net model, then it can be found [104], [38]. When an optimal net supervisor does not exist, the work in [104] and [38] does not offer a deadlock control solution. In this case, an interesting problem is to find a best permissive liveness-enforcing Petri net supervisor such that there are no other Petri net supervisors that are more permissive than it. The work in [14] presents a deadlock prevention approach to find a maximally permissive liveness-enforcing supervisor for an AMS if such a supervisor exists. Otherwise, it can derive a best permissive liveness-enforcing Petri net supervisor in the sense that there do not exist other Petri net supervisors that are more permissive than it.

Chen *et al.* [13], [14], [15] develop a novel method that can definitely find an optimal supervisor by adding monitors if such a supervisor exists. This method aims to block the uncontrolled system from entering into the deadlock-zone by preventing all the first-met bad markings (FBM) from being reached. Moreover, they formulate a method to ensure that all legal markings can be reached in the controlled system and a technique to reduce the computation burden by considering only a minimal set of legal markings and a minimal set of FBM via a vector covering approach.

The work in [14] suffers from the structural complexity problem since the number of the computed control places is not minimal. In [13], they propose an approach that can obtain a maximally permissive liveness-enforcing supervisor with the minimal number of control places. It is a

non-iterative approach since all control places can be once obtained by solving an integer linear programming problem (ILPP) (denoted as MCPP in [13]). Though this approach overcomes the problems of both behavior permissiveness and structural complexity, it still suffers from expensive computational cost.

The work in [15] employs a small (not minimal) number of monitors but more efficient by overcoming the computational complexity problem in [13]. They reduce the number of monitors by solving an ILPP at each iteration, when a place invariant for a control place is constructed to forbid FBM as many as possible and to allow the reachability of all markings in the minimal covering set of legal markings. This is achieved by maximizing the number of FBM forbidden by a place invariant (PI) via the objective function of the ILPP. By removing the forbidden FBM from the minimal covered set of FBM, this process is repeated until all FBM are forbidden.

These contributions, though correct and sound, are however far from being the cutting-edge of the literature, given that a better ILP model (plus more or less the same greedy heuristic) has been proposed in [86]. This model has a number of constraints which is linear with respect to the number of states, as opposed to the quadratic one required by the model in [13]. If the quadratic model were tighter than the linear one, that would compensate for its larger size. However, the experimental results which can be obtained implementing both models on an ILP solver and comparing their performance on benchmark instances, point towards a negative answer.

What is more, the ILP models commonly adopted in the literature have theoretical weaknesses, as discussed in [20], where it is shown how several large benchmark instances can be solved in a matter of seconds to guarantee optimality by an ad hoc algorithm, whereas an ILP solver requires hours, and often must be terminated without achieving an optimality guarantee, not to mention the recent extension of this theory to more complex supervisory control structures presented in [87]. All theses approaches require reachability analysis and some computation to compute critical live and forbidden markings.

Deadlock prevention based on siphon control is a typical application of structural analysis techniques of Petri nets. Siphons, a structural object of a Petri net, are widely used to analyze the deadlock problems in Petri nets. Deadlock control by using siphons can avoid the state explosion problem. Researchers have developed a large number of deadlock control policies based on siphon control, among which the representative works are given in [23], [4], [16], [52], [89], [47], [69],

[103], [48], [9], [122], [90], and [91].

In [23], Ezpeleta *et al.* develop a design method of monitor-based liveness-enforcing Petri net supervisors for AMS. This seminal work is usually considered to be a classical contribution that utilizes structural analysis techniques of Petri nets to prevent deadlocks in AMS. For a typical class of ordinary Petri nets, systems of simple sequential processes with resources (S^3PR), the work in [23] proposes a deadlock prevention policy by adding a control place to each possibly emptiable strict minimal siphon (SMS) to prevent itself from being emptied. The significance of this approach is that it successfully separates a plant net model and its supervisor. However, it is time-consuming for a sizable plant model since the number of such siphons in a net grows very quickly and may grow exponentially with respect to its size [58]. Moreover the approach in [23] suffers from the following problems : behavioral permissiveness, computational complexity, and structural complexity.

Due to the inherent complexity of Petri nets, any deadlock prevention policy that depends on a complete siphon enumeration is definitely exponential with respect to the size of its plant net model. In [16], Chu and Xie first use mixed integer programming (MIP) to detect whether a structurally bounded Petri net is deadlock-free. This method avoids the explicit enumeration of all strict minimal siphons and opens a new research avenue. Specifically, given a Petri net, a maximal unmarked siphon can be obtained by the following traditional siphon solution. First, remove all the unmarked places. Then remove the transitions without input places as well as their output places. Repeat the two steps until no places and transitions can be removed. A feasible solution corresponds to a maximal unmarked siphon when there exists a siphon that can be emptied at a marking that is reachable from the initial marking. Otherwise, its optimal solution is equal to the number of all the places in the Petri net. Although an MIP problem is NP-hard in theory [111], extensive numerical studies show that its computational efficiency is relatively insensitive to the initial marking and is more efficient than those that depend on a complete state or siphon enumeration. Deadlock control is usually concerned with minimal siphons. Huang *et al.* [47] propose an iterative two-stage deadlock prevention policy based on the work in [16]. At each iteration, an unmarked maximal siphon is detected by solving an MIP problem. If such a siphon exists, then an algorithm extracts a strict minimal siphon from the maximal one. In [7], [36], and [11], the MIP technique is also used. Their methods can find a minimal unmarked siphon directly.

A liveness-enforcing supervisor based on a complete siphon enumeration technique suffers from high structural complexity when the number of siphons is large. This problem has been recognized for many years. By fully utilizing the topological structure of a Petri net, the concepts of elementary and dependent siphons in a Petri net are proposed by Li and Zhou [69], [70]. They claim that siphons in a Petri net can be divided into elementary and dependent ones. The latter can be further distinguished by strongly and weakly dependent siphons with respect to elementary ones. It is shown that the number of the elementary siphons in a net is bounded by the smaller of place and transition counts. In many cases, monitors can be added for elementary siphons only. The controllability of a dependent siphon can be ensured by properly supervising the initial number of tokens in the monitors that are added to its elementary siphons. That is to say, a dependent siphon can be implicitly controlled by controlling its correlative elementary siphons. This is fully illustrated in [69] by an AMS example. The major contribution of the elementary theory is that it does lower the structural complexity of the supervisor notably. Note that the method in [69] does not lower the computational complexity or improve the behavior permissiveness compared with the policy in [23].

In an ordinary Petri net, a siphon is said to be controlled if it cannot be unmarked at any reachable marking [23], [46], [69]. If a Petri net is generalized, however, the controllability of a siphon is much more complex. Owing to the weights of arcs, the non-emptiness of a siphon is not sufficient for the absence of dead transitions. The existence of a strict minimal siphon is no longer necessary for the occurrence of deadlocks. As a whole, the controllability concept is concerned with the enabling and firing of transitions.

As a typical class of generalized Petri nets, a system of sequential systems with shared resources (S^4R) is proposed in [4]. It can model more complicated resource allocation systems with multiple concurrent processes. Different types of multiple resources can be requested by different processes. Thus, an S^4R has better modeling power than an S^3PR that is composed of states machines and resources [23]. Hence, solving a siphon control problem for S^4R assumes significance in designing liveness-enforcing supervisors.

However, the weight of an arc in a generalized Petri net can be an arbitrarily positive integer such that it is difficult to properly decide the lower bound of the number of tokens in a siphon. Motivated by this notorious issue, researchers propose a number of concepts involving the control-

lability of siphons in a generalized Petri net, such as max-controllability [4], max′-controllability [9], [9], [126], and max′′-controllability [80]. The proposal of these concepts aims to reduce the conservativeness[1] of a deadlock prevention policy whose development is based on the siphon control. Accordingly, a number of sufficient but not necessary liveness conditions are developed. This motivates us to find a more general controllability condition of siphons in generalized Petri nets.

Generalized system of simple sequential processes with resources (GS^3PR) is a subclass of S^4R and a generalized version of an S^3PR. It is easy to understand that the decision conditions for S^4R still hold for GS^3PR. The research on the necessary and sufficient condition for the siphon control in GS^3PR will be an important progress in deadlock control of generalized Petri nets.

Siphons are well recognized to be tied with deadlocks, which is true in both ordinary and generalized Petri nets. Iterative deadlock control is a classic strategy in deadlock prevention. Tricas utilizes an iteration approach to prevent deadlocks for AMS [100], [101], [102], [103]. At each iteration step, a siphon is computed and controlled by a monitor. Such a process is continued until all siphons are controlled. For an S^4R, this class of iterative deadlock prevention policies is usually believed to converge at some step although it is not an easy job to provide a formal yet satisfactory proof. The work has the advantage of avoiding the state explosion problem. However, such an iterative approach, in a general case, hardly leads to an optimal supervisor due to the unmature siphon control techniques for generalized Petri nets if deadlocks are eliminated by means of the concepts of max-controlled [4] or max'-controlled [9] siphons.

In [124], based on deadly marked siphons (DMS) [89] in well-marked S^4R, Zhao *et al.* modify the MIP test in [16] to detect DMS for S^4R. However, an S^4R may have livelocks even though it is deadlock-free. In this case, the siphons causing livelocks cannot be detected by the modified MIP and the net cannot be further controlled. Furthermore, the techniques in both [16] and [124] cannot obtain a minimal problematic siphon directly.

In [125], Zhong *et al.* propose an MIP model to detect a minimal non-max-marked siphon [125]. However, their method cannot detect the siphons that cause livelocks. Furthermore, it outputs an SMS when a Petri net is live with non-max-marked siphons, creating a false impression

[1] A deadlock prevention policy is said to be *conservative* if its resulting supervisor excludes some legal states, which implies that the supervisor is not maximally permissive.

that the net is non-live and thus needs a control place to control it.

In [81], the existing MIP-based methods are improved in the literature in terms of the max′′-controllability condition of siphons. We define extended DMS (EDMS) and then develop a more general MIP model that can detect deadlocks and livelocks caused by siphons in an S^4R. We conclude that the net is live if there is no feasible solution for the MIP model. This programming is more powerful than the MIPs in [124] and [125] but still restrictive since it outputs an SMS when a Petri net is live with non-max′′-marked siphons.

Recently, several deadlock control policies based on combination of state space and structural analysis have been proposed. The work in [112] can be considered as an improvement of the theory of regions. It designs a supervisor for a plant net model with maximally permissive behavior by using the theory of regions. Then, the SMS in the maximally permissive controlled system are computed and divided into elementary and dependent ones. To prevent them from being emptied, algebraic expressions about the markings of the additional monitors in the supervisor and the resource places in the plant net model are derived, under which the supervisor is live. The expressions are used to derive the live initial markings for the supervisor without changing its structure when the initial marking of the plant changes. A case study shows that the combined method is computationally efficient compared to existing ones in which the theory of regions is used alone, and the permissive behavior of the supervisor is near-optimal.

In [90], Piroddi *et al.* point out that there are several important drawbacks in the deadlock prevention methods that are based on elementary siphons [69], [75]. Firstly, elementary siphons are developed by purely utilizing the topological structure of a net, not taking into account of the dynamical evolution information of the net. Secondly, the policies based on elementary siphons are generally not maximally permissive, since the controlled siphons may be constrained to keep more than one token. Thirdly, the set of elementary siphons in a Petri net is not unique. The existence of different sets of elementary siphons also implies that the deadlock prevention solution is not unique. Last but not least, the policies based on elementary siphons can be applied to some special classes of Petri nets only. Piroddi *et al.* believe that it is important to integrate the structural information related to strict minimal siphons with reachability graph analysis to avoid unnecessary control places. The work in [90] develops a selective siphon control policy in which the concepts of essential, dominated, and dominating siphons and critical, dominating and dominated markings

play an important role. By solving set covering problems, dominating siphons are found to ensure that dominated siphons are controlled. The resulting supervisor is highly permissive. The major technical problem in [90] is its computational complexity. At each iteration, it needs to compute all minimal siphons and all dominating markings and to solve a set covering problem, each of which is NP-hard in theory with respect to the net size. Later, in [91], Piroddi *et al.* improve the method by using the MIP-based deadlock detection approach such that the complete minimal siphon enumeration is avoided.

2.2 Automated Manufacturing Systems with Unreliable Resources

All the studies reviewed in the previous section assume that resources do not fail. Actually, resource failures are inevitable in most AMS, which may also cause an AMS to be deadlocked. Thus, it is a necessary requirement to develop an effective and robust deadlock control policy to ensure that deadlocks cannot occur even if some resources in a system break down.

There is a lack of research in Petri nets regarding the impacts of unreliable resources on AMS under the supervisory control of deadlocks. In fact, resource failures are a common problem in real-world systems, which pose challenges in supervisory control of discrete event systems including AMS. In case of resource failures, the existing deadlock control policies are always no longer in force and deadlocks in the disturbed system may be caused. Therefore, reanalysis of the disturbed system is usually necessary. Robustness analysis provides an alternative way to determine whether the operation of a disturbed system or a part of it can still be maintained in case of resource failures. To the best of our knowledge, no much work is found on robust supervision of AMS based on Petri nets.

Reveliotis [95] considers a scenario where parts requiring a failed resource can be rerouted or removed from a system through human intervention. Park and Lim [88] address the existence issues of robust supervisors. Lawley *et al.* [67], [97], [98] design supervisors for unstable systems based on the banker's algorithm and central buffer constraints with the following properties : (1) the supervisor ensures continuing production of part types, not requiring failed resources ; (2) the supervisor allows only those states that serve as feasible initial states if additional resource failure occurs ; and (3) the supervisor allows only those states that serve as feasible initial states if failed resources are repaired.

Hsieh develops a variety of methods to determine the feasibility of production with a set of resource failures modeled as the extraction of tokens from a Petri net [39], [40], [41], [42], [43], [44], [45]. This researcher has studied the robustness of several subclasses of Petri nets, including controlled production Petri nets (CPPN) [39], controlled assembly Petri nets (CAPN) [40], controlled assembly/disassembly Petri nets (CADPN) [41], controlled assembly Petri nets with alternative routes (CAPN-AR) [42], collaborative Petri nets (CPN) [43], and non-ordinary controlled flexible assembly Petri nets with uncertainties (NCFAPNU) [45]. In these works, liveness conditions and robustness analysis of the nets are based on the concepts of token flow paths and minimal resource requirements (MRR). His work reports fault tolerant conditions and proposes a structural decomposition method to test the feasibility of production routes. However, all these methods are not intuitive to the Petri net models. In this dissertation, we try to enforce liveness and robustness via a supervisor by adding monitors and recovery subnets. This implies that both a plant and its supervisor are unified in a Petri net formalism.

An interesting issue is how to make the existing deadlock control policies possess a desirable robust property to cope with resource failures. Specifically, the desirable robustness is a system property to keep a controlled system live as some resources break down. In this thesis, we focus on robust supervision of AMS. We hope that the supervisor designed for AMS with unreliable resources has following properties : (1) it can prevent deadlocks for a plant model when all resources work normally, (2) deadlocks are prevented even if some resources fail to work and be removed to repair at any time, and (3) deadlocks disappear after the repaired resources are returned.

3. Thesis Organization

This thesis is intended to provide deadlock resolution based on Petri nets and robust supervisor design for AMS. The proposed methods mostly rely on Petri net structural analysis. The thesis also deals with the cases of unreliable resources in AMS when designing deadlock controllers.

In Introduction, we first recall the AMS and discuss the importance of the deadlock problems and their resolution. Then, we review two different aspects of AMS, i.e., liveness-enforcing supervision for AMS and AMS with unreliable resources.

Chapter 1 introduces the necessary basics of Petri nets as well as the notations used throughout

this thesis.

Chapter 2 reviews the concepts of max, max′, max′′-controlled siphons and formulates a new concept called max*-controlled siphons for GS^{3}PR. We conclude that a GS^{3}PR is live iff all its strict minimal siphons are max*-controlled. Then examples are given to illustrate the max, max′, max′′, max*-controlled siphons and their difference. Compared with the existing ones, the proposed concept is more general. Also, some open problems are discussed. Based on the max*-controllability condition of siphons, we propose a new integer programming (IP) model that can detect minimal non-max*-marked siphons that cause deadlocks or livelocks directly. We conclude that if there is no feasible solution to this model, the net is live. Since the approach is based on siphons and mathematical programming, its computational efficiency is relatively insensitive to the initial marking. Compared with the existing methods, the proposed one is more powerful. Experimental studies are conducted to illustrate it.

Chapter 3 reports a novel design method of deadlock prevention supervisors based on Petri nets. It does not guarantee optimality but empirical results show its superiority over other approaches based on siphon control. Given the Petri net model of an AMS, one first designs an optimal liveness-enforcing controlled system for the model under a minimal initial marking by utilizing the theory of regions. Then, we calculate all SMS in the controlled system. Such a siphon does not contain a trap. For each SMS, an algebraic inequality with respect to the markings of monitors and resource places in the controlled system, also called a liveness constraint, is established in terms of the concept of max-controlled or invariant-controlled siphons. Its satisfaction implies the absence of dead transitions in the postset of the corresponding siphon. Consequently, given initial markings that satisfy all the liveness inequality constraints, all siphons can be max-controlled, and the resulting controlled system is live. After a controlled system structure is found, one can reallocate the initial markings according to the inequality constraints. No matter how large the initial markings and the number of states are, the liveness constraints remain unchanged. Their satisfaction ensures the absence of uncontrolled siphons.

Chapter 4 proposes a maximally permissive control policy for a subclass of S^{3}PR (called β-nets) based on the theory of token distribution pattern of siphons. We first show that by adding a monitor for each critical siphon, some live states may get lost since the monitor controls the complementary set of a critical siphon, where some places may not be marked. By controlling

only the set of marked operation places, more live states can be reached. However, this induces some emptiable siphons. The corresponding token pattern can be inferred. By adding monitors to all such possibly emptiable siphons, the controlled net becomes live and maximally permissive. There is no need to construct a reachability graph and enumerate all minimal siphons.

A variety of deadlock control policies based on Petri nets have been proposed for AMS. Most of them prevent deadlocks by adding monitors for emptiable siphons that, without an appropriate control policy, can cause deadlocks, where the resources in a system under consideration are assumed to be reliable. When resources are unreliable, it is difficult or impossible to apply existing control strategies. For S^3PR, Chapter 5 bridges the gap between a divide-and-conquer deadlock control strategy and its application to real-world systems with unreliable resources. Recovery subnets and monitors are designed for unreliable resources and strict minimal siphons that may be emptied, respectively. Normal and inhibitor arcs are used to connect monitors with recovery subnets in case of necessity. Then reanalysis of the original Petri net is avoided and a robust liveness-enforcing supervisor is derived. Examples are presented to illustrate the proposed methodology.

Finally, we summarize this dissertation by highlighting the major contributions. Limitations and future research issues are outlined. Several unsolved and interesting problems are illustrated.

Chapitre 1

Preliminaries of Petri Nets

1.1 Introduction

This chapter presents the formal definition of Petri nets and the related concepts, including structural and behavioral properties such as invariants, siphons, traps, liveness, and boundedness. A number of important subclasses of Petri nets are introduced. This chapter is fundamental for understanding of the ideas presented in the following chapters. For more details, please refer to [75], [132], [85], [6], and [83].

1.2 Formal Definitions

From the perspective of graph theory, a Petri net is a directed bipartite graph. It consists of two components : a net structure and an initial marking. A net structure is composed of two kinds of nodes, namely places and transitions, and directed arcs from places to transitions or from transitions to places. Graphically, places are represented by circles and transitions by boxes or bars. Tokens in a place can be denoted by black dots, or a positive integer representing their number. The initial token distribution is called the initial marking.

Definition 1.1 *A Petri net 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* $P \cup T \neq \emptyset$ *and* $P \cap T = \emptyset$. $F \subseteq (P \times T) \cup (T \times P)$ *is called a flow relation of the net, represented by arcs with arrows from places to transitions or from transitions to places.* $W : (P \times T) \cup (T \times P) \to \mathbb{N}$ *is a mapping that assigns a weight to an arc :* $W(f) > 0$ *if* $f \in F$ *and* $W(f) = 0$ *otherwise, where* $\mathbb{N} = \{0, 1, 2, \ldots\}$.

$N = (P, T, F, W)$ is ordinary, denoted as $N = (P, T, F)$, if $\forall f \in F$, $W(f) = 1$. It is said to be a generalized net if $\exists f \in F$, $W(f) > 1$.

Usually, a place represents a condition, a resource, or an activity while a transition represents an event. A token in a place means the fulfilment of a condition, the availability of a resource, or the process of an activity.

Definition 1.2 *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 $P_X \cup T_X$ if $F_X = F \cap [(P_X \times T_X) \cup (T_X \times P_X)]$ and $\forall f \in F_X$, $W_X(f) = W(f)$.*

Definition 1.3 *A marking M of a Petri net N assigns to each place a nonnegative integer. To facilitate linear algebraic analysis, a marking M is usually treated as a $|P|$-vector. $M(p)$ denotes the number of tokens in place p. For economy of space, $\sum_{p \in P} M(p)p$ is used to denote vector M. Place p is marked at M if $M(p) > 0$. Given a subset $S \subseteq P$, the sum of tokens in all the places in S is denoted by $M(S)$ with $M(S) = \sum_{p \in S} M(p)$. S is marked (unmarked) at M if $M(S) > 0$ ($M(S) = 0$). Let M_0 be an initial marking of net N. (N, M_0) is called a marked net.*

Exemple 1.1 *As shown in Fig. 1.1, (N, M_0) is a Petri net with $P = \{p_1 - p_8\}$, $T = \{t_1 - t_6\}$, $F = \{(p_1, t_1), (t_1, p_2), (p_2, t_2), (t_2, p_3), (p_3, t_3), (t_3, p_1), (p_4, t_4), (t_4, p_5), (p_5, t_5), (t_5, p_6), (p_6, t_6), (p_7, t_1), (p_7, t_2), (p_7, t_5), (t_3, p_7), (t_6, p_7), (p_8, t_2), (p_8, t_4), (t_3, p_8), (t_5, p_8)\}$, $W(p_1, t_1) = W(t_1, p_2) = W(p_2, t_2) = W(t_2, p_3) = W(p_3, t_3) = W(t_3, p_1) = W(p_4, t_4) = W(t_4, p_5) = W(p_5, t_5) = W(t_5, p_6) = W(p_6, t_6) = W(t_6, p_4) = W(p_7, t_1) = W(p_8, t_2) = W(p_8, t_4) = W(t_3, p_8) = W(t_5, p_8) = 1$, $W(p_7, t_2) = W(p_7, t_5) = W(t_6, p_7) = 2$, and $W(t_3, p_7) = 3$. It is clear that the net is a generalized Petri net. Places p_1 has three tokens, denoted by three black dots in this figure. The initial marking of this Petri net can be denoted as $M_0 = 3p_1 + 2p_4 + 3p_7 + p_8$.*

Definition 1.4 *A net $N = (P, T, F, W)$ is pure (self-loop free) if $\forall x, y \in P \cup T$, $W(x, y) > 0$ implies $W(y, x) = 0$. A pure net structure can be fully described by its incidence matrix denoted by $[N]$ that is a $|P| \times |T|$ integer matrix with $[N](p, t) = W(t, p) - W(p, t)$. Alternatively, $[N]$ can be represented by Post−Pre, where $Pre : P \times T \to \mathbb{N}$ and $Post : P \times T \to \mathbb{N}$.*

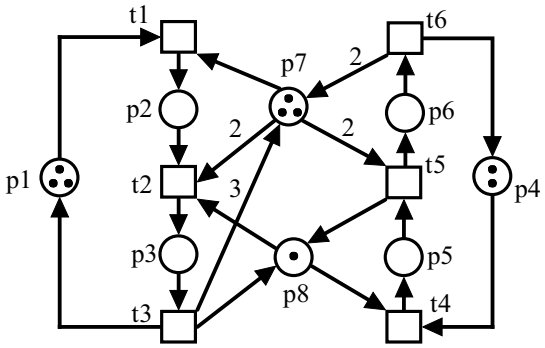


FIG. 1.1 – A Petri net (N, M_0).

Exemple 1.2 *For the Petri net shown in Fig. 1.1, its post-incidence matrix, pre-incidence matrix, and incidence matrix are as follows :*

$$\text{Post} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 3 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}, \text{Pre} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 2 & 0 & 0 & 2 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$[\text{N}] = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ -1 & -2 & 3 & 0 & -2 & 2 \\ 0 & -1 & 1 & -1 & 1 & 0 \end{bmatrix}.$$

Definition 1.5 *Let $N = (P, T, F, W)$ be a net and σ a finite sequence of transitions. The Parikh vector of σ is $\overrightarrow{\sigma} : T \to \mathbb{N}$ which maps t in T to the number of occurrences of t in σ. Denote $\overrightarrow{t_1} = (1, 0, \cdots, 0)^T$, $\overrightarrow{t_2} = (0, 1, 0, \cdots, 0)^T$, and $\overrightarrow{t_k} = (0, 0, \cdots, 0, 1)^T$ assuming $k = |T|$.*

It is trivial that for each transition t, we have $[N](\cdot, t) = [N]\overrightarrow{t}$. Note that $M[t\rangle M'$ leads to $M' = M + [N](\cdot, t)$. Consequently, if $M[t\rangle M'$, we have $M' = M + [N]\overrightarrow{t}$. For an arbitrary finite

transition sequence σ such that $M[\sigma\rangle M'$, we have

$$M' = M + [N]\overrightarrow{\sigma}$$

This equation is called the state equation of a Petri net (N, M), which presents an algebraic description of the marking change in a Petri net. Such a linear algebraic expression is very helpful because it allows one to apply the concepts and results of linear algebra to the domain of Petri nets.

Definition 1.6 *Given a node $x \in P \cup T$, ${}^\bullet x = \{y \in P \cup T \mid (y, x) \in F\}$ is called the preset of x, while $x^\bullet = \{y \in P \cup T \mid (x, y) \in F\}$ is called its postset. We can extend this notation to a set of nodes as follows : given $S \subseteq P \cup T$, ${}^\bullet S = \cup_{x \in S} {}^\bullet x$ and $S^\bullet = \cup_{x \in S} x^\bullet$.*

For $p \in P$, $t \in {}^\bullet p$ and $t \in p^\bullet$ are called an input and an output transition of p, respectively. For $t \in T$, $p \in {}^\bullet t$ and $p \in t^\bullet$ are called an input and an output place of t, respectively.

Exemple 1.3 *In Fig. 1.1, ${}^\bullet t_1 = \{p_1, p_7\}$, $t_1^\bullet = \{p_2\}$, ${}^\bullet p_3 = \{t_2\}$, ${}^\bullet p_6 = \{t_5\}$, ${}^\bullet p_7 = \{t_3, t_6\}$, $p_3^\bullet = \{t_3\}$, $p_6^\bullet = \{t_6\}$, and $p_7^\bullet = \{t_1, t_2, t_5\}$. Given $S = \{p_3, p_6, p_7\}$, ${}^\bullet S = {}^\bullet p_3 \cup {}^\bullet p_6 \cup {}^\bullet p_7 = \{t_2, t_3, t_5, t_6\}$, and $S^\bullet = p_3^\bullet \cup p_6^\bullet \cup p_7^\bullet = \{t_1, t_2, t_3, t_5, t_6\}$.*

Definition 1.7 *Given a marking M, an arc $(p, t) \in F$ is said to be enabled (disabled) if $M(p) \geq W(p, t)$ ($M(p) < W(p, t)$). For an arc (p, t), e_{pt}, a binary variable, indicates whether the arc is enabled. $e_{pt} = 1$ ($e_{pt} = 0$) means that the arc is enabled (disabled).*

Definition 1.8 *In net (N, M_0), A transition t is enabled (disabled) at M if $\forall p \in {}^\bullet t$ ($\exists p \in {}^\bullet t$), arc (p, t) is enabled (disabled). This fact can be denoted by $M[t\rangle$. Firing it can reach a new marking M' with $M'(p) = M(p) - W(p, t) + W(t, p)$, denoted by $M[t\rangle M'$. Marking M' is said to be reachable from M if there exist a sequence of transitions $\sigma = t_0 t_1 \ldots t_n$ and markings M_1, M_2, ..., and M_n such that $M[t_0\rangle M_1[t_1\rangle \ldots M_n[t_n\rangle M'$ holds. The set of markings reachable from M in N is called the reachability set of Petri net (N, M) and denoted as $R(N, M)$. Petri net is reversible if $\forall M \in R(N, M_0)$, $M_0 \in R(N, M)$.*

Definition 1.9 *Given a marked Petri net (N, M_0), a transition $t \in T$ is live at M_0 if $\forall M \in R(N, M_0)$, $\exists M' \in R(N, M)$, $M'[t\rangle$ holds. A transition $t \in T$ is said to be dead at marking $M \in R(N, M_0)$, if $\nexists M' \in R(N, M)$, $M'[t\rangle$. (N, M_0) is live at M_0 if $\forall t \in T$, t is live at M_0. Otherwise, (N, M_0) is*

non-live. (N, M_0) *is deadlocked at* M *if* $\nexists t \in T,\ M[t\rangle$*, where* $M \in R(N, M_0)$ *and* M *is called a dead marking.* (N, M_0) *is deadlock-free if* $\forall M \in R(N, M_0),\ \exists t \in T,\ M[t\rangle$.

Definition 1.10 *A marked net is bounded if* $\exists k \in \mathbb{N},\ \forall M \in R(N, M_0),\ \forall p \in P, M(p) \leq k$. *A net* N *is structurally bounded if it is bounded for any initial marking.*

Exemple 1.4 *For the Petri net in Fig. 1.1,* t_1 *and* t_4 *are enabled at* $M_0 = 3p_1 + 2p_4 + 3p_7 + p_8$ *since* $^\bullet t_1 = \{p_1, p_7\}$, $M_0(p_1) = 3 > W(p_1, t_1) = 1$, $M_0(p_7) = 3 > W(p_7, t_1) = 2$, $^\bullet t_4 = \{p_4, p_8\}$, $M_0(p_4) = 2 > W(p_4, t_4) = 1$, *and* $M_0(p_8) = W(p_6, t_4) = 1$. *Firing* t_1 *at* M_0 *leads to* $M_1 = 2p_1 + p_2 + 2p_4 + 2p_7 + p_8$. *Firing* t_4 *at* M_0 *leads to* $M_2 = 3p_1 + p_4 + p_5 + 3p_7$. *The reachability set of the net is* $R(N, M_0) = \{M_0, M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{12}\}$, *where* $M_0 = 3p_1 + 2p_4 + 3p_7 + p_8$, $M_1 = 2p_1 + p_2 + 2p_4 + 2p_7 + p_8$, $M_2 = p_1 + 2p_2 + 2_4 + p_7 + p_8$, $M_3 = 3p_2 + 2p_4 + p_8$, $M_4 = 3p_2 + p_4 + p_5$, $M_5 = p_1 + 2p_2 + p_4 + p_5 + p_7$, $M_6 = 2p_2 + p_3 + 2p_4$, $M_7 = 2p_1 + p_2 + p_4 + p_5 + 2p_7$, $M_8 = 2p_1 + p_2 + p_4 + p_6 + p_8$, $M_9 = 2p_1 + p_2 + p_5 + p_6$, $M_{10} = 3p_1 + p_4 + p_5 + 3p_7$, $M_{11} = 3p_1 + p_4 + p_6 + p_7 + p_8$, $M_{12} = 3p_1 + p_5 + p_6 + p_7$. *Fig. 1.2 shows its reachability graph. The net is non-live and bounded.*

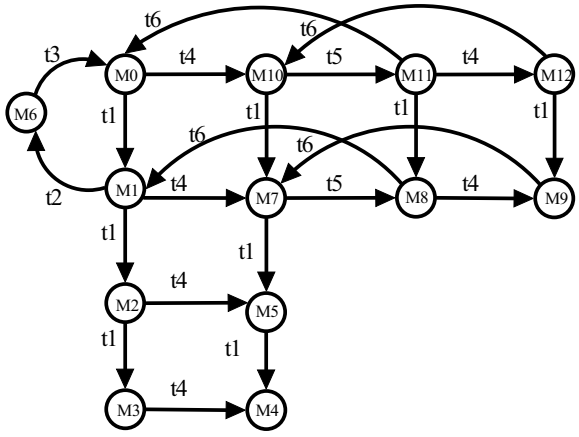


FIG. 1.2 – The reachability graph of net (N, M_0) shown in Fig. 1.1.

Definition 1.11 *Let* x *be a node in a Petri net* $N = (P, T, F, W)$ *and* (x_i, x_{i+1}) *be a directed arc from node* x_i *to* x_{i+1}. *A sequence* $x_0, (x_0, x_1), x_1, \ldots, x_{n-1}, (x_{n-1}, x_n), x_n$ *is called a directed path of* N *if* $\forall x \in \{x_0, \ldots, x_n\},\ x \in P \cup T$ *and* $\forall i \in \{1, 2, \ldots, n\},\ (x_{i-1}, x_i) \in F$. *An elementary path from* x_0 *to* x_n *is a path whose nodes are all different (except, perhaps,* x_0 *and* x_n*). An elementary path is denoted*

by $EP(x_0, x_n)$. A circuit is an elementary path with $x_0 = x_n$. Petri net N is strongly connected if there is a directed path from each node x to every other node y, where $x, y \in P \cup T$.

Definition 1.12 *A net $N = (P, T, F)$ is called a state machine if $\forall t \in T$, $|{}^\bullet t| = |t^\bullet| = 1$.*

1.3 Deadlocks and Livelocks

In order to facilitate understanding and readability, this section gives some visual interpretation for deadlocks and livelocks by using reachability graphs.

A net system is bounded if its reachability set has a finite number of elements. The reachability set of net (N, M_0) can be expressed by a reachability graph that is a directed graph whose nodes are markings in $R(N, M_0)$ and arcs are labeled by the transitions of N. An arc from M_1 to M_2 is labeled by t if $M_1[t\rangle M_2$.

In the reachability graph of a Petri net, a global deadlock is a terminal node that corresponds to system states from which the system cannot further evolve. While the existence of local deadlocks is referred to as livelocks. When a system is at a livelock state, the overall states of the system continues to change while parts of the system are deadlocked [57]. Actually, livelock is a special case of resource starvation although the system may be deadlock-free. Formally, if a system has livelock states, its reachability graph must contain a strongly connected component with two or more nodes and without outgoing arcs (no exit to leave the component).

Fig. 1.2 shows the reachability graph of the net (N, M_0) shown in Fig. 1.1. Marking M_4 is a global deadlock. When the system is at this state, it cannot go back to the initial state. This state must be avoided when a deadlock control policy is designed.

Take the reachability graph of the Petri net model shown in Fig. 1.3 as an example. By using INA [50], we can obtain 60 reachable states, 56 of which are permissive behavior. This net is deadlock-free. Here, we only show the livelock part due to its large size. In Fig. 1.4, the components in the dashed line are at livelocks. For states $M_{36} = p_1 + p_3 + p_5 + p_8 + 5p_9 + 5p_{10} + 5p_{11}$, $M_{37} = p_2 + p_3 + p_5 + p_7 + 5p_9 + 5p_{10} + 5p_{11}$, $M_{38} = p_1 + p_2 + p_3 + p_5 + 4p_9 + 5p_{10} + 5p_{11}$, and $M_{39} = p_3 + p_5 + p_7 + p_8 + 6p_9 + 5p_{10} + 5p_{11}$, there are no outgoing arcs to leave the component in the dashed-line box.

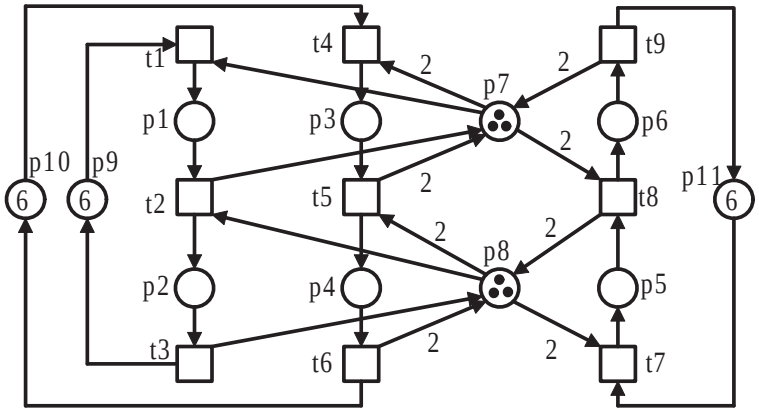


FIG. 1.3 – A Petri net (N, M_0) in [81].

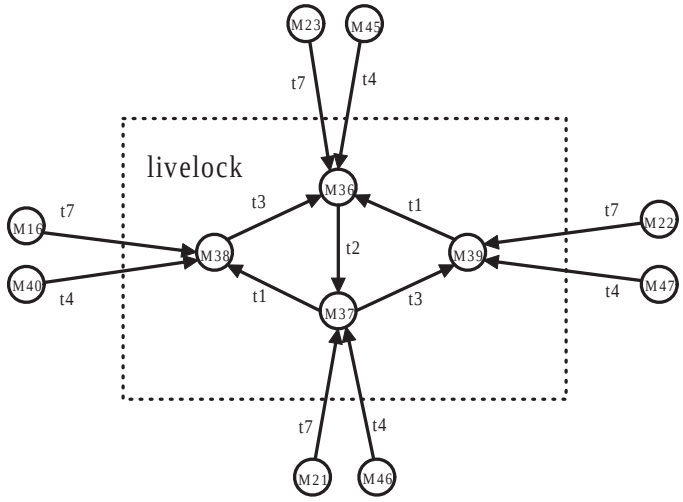


FIG. 1.4 – An example of livelocks shown in a reachability graph.

Livelocks must be considered when solving deadlock control problems since once a system is at livelock, other processes cannot be processed smoothly. Broadly speaking, livelock is a special case of deadlocks. It is very significant to find a mechanism to detect deadlocks and livelocks in a system.

1.4 Inhibitor Arc

The modeling power of Petri nets can be increased by inhibitor arcs. An inhibitor arc, denoted as $(p, t)^o$, connects an input place and a transition, and is graphically represented by an arc from a place p to a transition t terminated with a small circle. The presence of an inhibitor arc changes the transition enabling conditions. A transition is regarded as enabled if each input place, connected to the transition by a normal arc (an arc terminated with an arrow), contains at least the number of tokens equal to the weight of the arc, and no tokens are present in each input place connected to the transition via an inhibitor arc. The transition firing rules are the same as for normally connected places. The firing, however, does not change the marking of the places connecting the transition by an inhibitor arc [134], [85]. We use ${}^\bullet t^o$ to denote the set of places from which there are inhibitor arcs to transition t. While $p^{o\bullet}$ denotes the set of transitions to which there are inhibitor arcs from place p.

A Petri net with an inhibitor arc $(p_2, t)^o$ is shown in Fig. 1.5. We have ${}^\bullet t^o = \{p_2\}$ and $p_2^{o\bullet} = \{t\}$. In this figure, t can fire at the current state. Firing t leads to $M(p_1) = M(p_2) = 0$ and $M(p_3) = 1$.

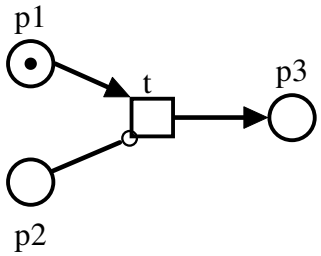


FIG. 1.5 – An extended Petri net with an inhibitor arc.

1.5 Structural Invariants, Siphons, and Traps

The structural properties that depend on only the topological structure of a Petri net and are independent of the initial marking are called invariants. Structural invariants are an important method to analyze the behavior of a Petri net from a structural viewpoint.

Definition 1.13 *A P-vector is a column vector $I : P \to \mathbb{Z}$ indexed by P and a T-vector is a column vector $J : T \to \mathbb{Z}$ indexed by T, where $\mathbb{Z}$ is the set of integers.*

P-vector I is denoted by $\sum_{p \in P} I(p)p$ for economy of space. For example, $I = (3, 1, 0, 0, 4)^T$ can be written as $I = 3p_1 + p_2 + 4p_5$. We denote a column vector whose entries equal 0(1) by $\mathbf{0}(\mathbf{1})$. I^T and $[N]^T$ are their transposed versions of I and $[N]$, respectively.

Definition 1.14 *P-vector I is called a P-invariant (place invariant) if $I \neq \mathbf{0}$ and $I^T[N] = \mathbf{0}^T$. A P-semiflow I is a P-invariant if every element of I is non-negative. $\|I\| = \{p \in P | I(p) \neq 0\}$ is called the support of I. $\|I\|^+ = \{p | I(p) > 0\}$ denotes the positive support of P-invariant I, while $\|I\|^- = \{p | I(p) < 0\}$ denotes the negative support of I. I is said to be a minimal P-invariant if there does not exist a P-invariant I' such that $\|I'\| \subset \|I\|$ and its components are mutually prime.*

P-invariants that can be derived from the state equation of a Petri net are marking invariants. Token count in their corresponding places keeps constant, i.e., the invariant law associated with a P-invariant holds for any reachable marking. Specifically, if I is a P-invariant of (N, M_0), then $\forall M \in R(N, M_0), I^T M = I^T M_0$.

Exemple 1.5 *Consider the Petri net in Fig. 1.1. There are four minimal P-invariants, i.e., $I_1 = p_1 + p_2 + p_3$, $I_2 = p_4 + p_5 + p_6$, $I_3 = p_2 + 3p_3 + 2p_6 + p_7$, and $I_4 = p_3 + p_5 + p_8$.*

Let X be a matrix where each column is a P-semiflow of (N, M_0). The set of invariant markings is denoted as $I_X(N, M_0) = \{M \in \mathbb{N}^m | X^T M = X^T M_0\}$, where m is the number of places of (N, M_0) and $\mathbb{N}^m$ indicates the set of m-dimensional non-negative integer vectors.

Similar to structural invariants, siphons and traps are also structural objects and play an important role in the analysis of Petri nets, particularly their liveness property. The combination of P-invariants and siphons can be used to design liveness-enforcing supervisors for Petri nets. A siphon remains empty once it loses all tokens while a trap remains marked once it has any token.

Definition 1.15 *A nonempty set $S \subseteq P$ is a siphon if ${}^\bullet S \subseteq S^\bullet$. $S \subseteq P$ is a trap if $S^\bullet \subseteq {}^\bullet S$. A siphon that does not contain the support of any P-semiflow is called a strict siphon. A strict siphon is called a strict minimal siphon (SMS) if there is no siphon contained in it as a proper subset.*

Definition 1.16 *A siphon S is said to be controlled in an ordinary Petri net system (N, M_0) if $\forall M \in R(N, M_0)$, $M(S) > 0$.*

When we talk about siphon control, we usually consider minimal siphons since the controllability of a minimal siphon implies that of those containing it.

Due to the definition of siphons, all transitions connected to a siphon can never be enabled once it is emptied. The transitions are therefore dead, leading to the fact that the net containing these transitions is not live. As a result, deadlock-freedom and liveness of a Petri net are closely related to its siphons, which is shown by the following known results [21].

Property 1.1

Let $S \subseteq P$ be a siphon of an ordinary net N. If S is controlled by a P-invariant I under M_0, S cannot be emptied, i.e., $\forall M \in R(N, M_0)$, S is marked at M.

Property 1.2

If an ordinary net system (N, M_0) is dead, the set of all unmarked places forms a siphon. If no minimal siphon in N can be emptied, (N, M_0) is deadlock-free.

Theorem 1.1

Let (N, M_0) be an ordinary net and Π the set of its siphons. The net is deadlock-free if $\forall S \in \Pi$, $\forall M \in R(N, M_0)$, $M(S) > 0$.

This theorem states that an ordinary Petri net is deadlock-free if no (minimal) siphon eventually becomes empty.

Theorem 1.2

Let (N, M) be an ordinary net that is in a deadlock state. Then, $S = \{p \in P | M(p) = 0\}$ is a siphon.

This result means that if an ordinary net is dead, i.e., no transition is enabled, then the unmarked places form a siphon.

Exemple 1.6 *In the net shown in Fig. 1.1,* $S = \{p_3, p_6, p_7\}$ *is an SMS since* ${}^\bullet S = \{t_2, t_3, t_5, t_6\}$, $S^\bullet = \{t_1, t_2, t_3, t_5, t_6\}$, *and* ${}^\bullet S \subset S^\bullet$. *From the reachability graph of the net, we can see that the net is not live.*

If a Petri net is generalized, however, the controllability of a siphon is much more complex. Owing to the weights of arcs, the non-emptiness of a siphon is not sufficient for the absence of dead transitions. The existence of an SMS is no longer necessary for the occurrence of deadlocks. As a whole, the controllability concept is concerned with the enabling and firing of transitions.

1.6 Subclasses of Petri Nets

Many results are applicable to certain classes of Petri nets only. Various net classes are proposed for AMS in literature. Different net classes may model different complex processes of AMS. In this section, we first review a necessary background in the place classification of Petri net models of flexible manufacturing systems. Then S^3PR, GS^3PR and S^4R nets are given to serve for latter chapters.

Let us consider a small manufacturing system consisting of a machine, a robot, an input buffer and an output buffer. The robot picks up a raw part from the input buffer, uploads the machine tool, and downloads the finished part from the machine and puts it into the output buffer after the machine finishes the operation on a raw part. Fig. 1.6(a) and 1.6(b) show the system and its Petri net model, respectively. In the Petri net model, place p_1 models the input and output buffers. $M_0(p_1) = 2$ means that at the initial state, there are two raw parts in the input buffer. Place p_5 models the machine. $M_0(p_5) = 2$ represents its processing capacity, indicating the two parts can be processed in it at the same time. Place p_6 models the robot. $M_0(p_6) = 1$ means that the robot can hold one part only at a time. Places p_2, p_3, p_4 model the operations of uploading by the robot, processing by the machine, and downloading by the robot, respectively. The machine and the robot are called fixed resources in the sense that their capacities are normally fixed. The input and output buffers are called variable resources since they can carry the variable number of raw parts.

In a Petri net model of a manufacturing system, a place represents either an operation or a resource status and a transition represents the start or end of an operation. Such a modeling paradigm can be traced back to the seminal works [127], [128], [129], and [23].

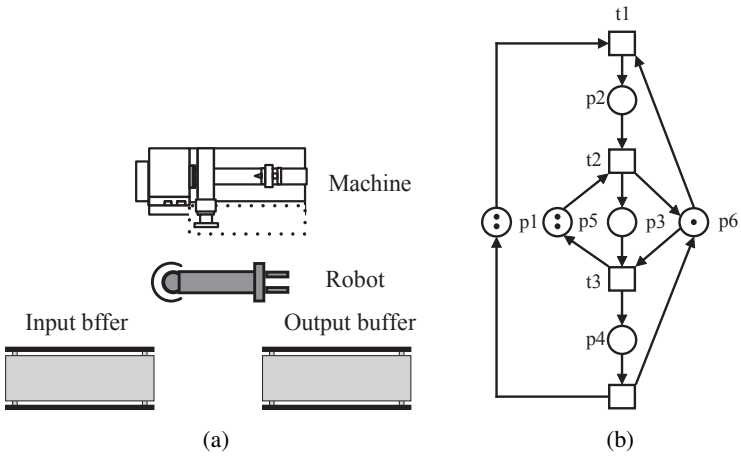


FIG. 1.6 – (a) A manufacturing system, (b) its net model (N, M_0)

Accordingly, the places in a Petri net model of a manufacturing system have the following partition : A-places, B-places, and C-places, depending on the physical characteristics of its resources, represented by the initial marking M_0 [127], [128], [129], which is adopted by the researchers in this area nowadays.

A-place : A place p is called an A-place or an activity place (also called an operation place) if $M_0(p) = 0$.

B-place : A place p is called a B-place or a fixed resource place if $M_0(p)$ is a constant.

C-place : A place p is called a C-place or a variable resource place if p is initially marked with $M_0(p) > 0$ and the number of initial tokens in p is variable, i.e., $M_0(p)$ is variable.

An A-place p often represents an activity place. Initially there is no operation in a system. A B-place p represents the availability of a fixed number of resources for a given system. A C-place p can represent the availability of raw parts. B-places and C-places model the availability of resources that are necessary to start some operations at certain stages. This classification has been well accepted by academic and industrial communities. Later, with the same partitioning rules, Ezpeleta et al. [23] rename A-places, B-places, and C-places to be activity places, resource places, and process idle places whose sets are denoted by P_A, P_R, and P^0, respectively.

In Fig. 1.6(b), p_1 is a process idle place. p_2, p_3, and p_4 are activity places, and p_5 and p_6 are resource places. That is to say, $P_A = \{p_2, p_3, p_4\}$, $P^0 = \{p_1\}$, and $P_R = \{p_5, p_6\}$. Suppose

that the nets in Fig. 1.7 are the models of manufacturing systems. In Fig. 1.7(a), we have $P_A = \{p_1, p_2, p_4, p_5\}$, $P^0 = \{p_3, p_6\}$, and $P_R = \{p_7, p_8\}$. In Fig. 1.7(b), we have $P_A = \{p_1, p_2, p_4, p_5\}$, $P^0 = \{p_3, p_6\}$, and $P_R = \{p_7, p_8\}$. In Fig. 1.7(c), we have $P_A = \{p_2, p_3, p_4, p_6, p_7\}$, $P^0 = \{p_1, p_5\}$, and $P_R = \{p_8, p_9\}$.

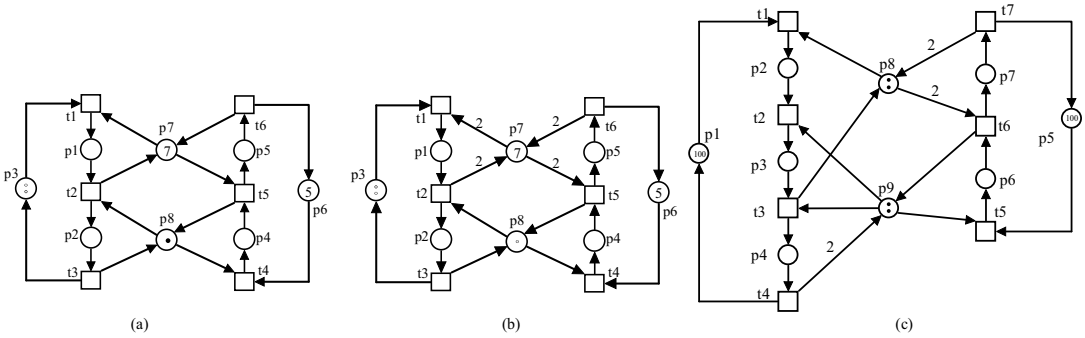


FIG. 1.7 – (a) An S^3PR, (b) a GS^3PR, (c) an S^4R.

The number of tokens in a process idle place indicates the maximal number of concurrent activities that can occur in a process. A token in an activity place means that a raw material is being processed and the tokens in a resource place represent its capacity indicating how many processing units can be provided by the resource type. For example, the operation modeled with place p_1 in Fig. 1.7(a) needs one unit of only one resource type, i.e., p_7. The activity place modeled with p_1 in Fig. 1.7(b) needs two units of only one resource type, i.e., p_7. The activity place modeled with p_3 in Fig. 1.7(c) needs one unit of resource p_8 and one unit of resource p_9.

1.6.1 S^3PR Net

Definition 1.17 *[75] A simple sequential Process (S^2P) is a Petri net $N = (P_A \cup \{p^0\}, T, F)$ where 1) $P_A \neq \emptyset$ is called the set of activity places ; 2) $p^0 \notin P_A$ is called the idle process place ; 3) N is a strongly connected state machine ; and 4) every circuit of N contains place p^0.*

Definition 1.18 *[75] A simple sequential process with resources (S^2PR) is a Petri net $N = (\{p^0\} \cup P_A \cup P_R, T, F)$ such that*

1. *The subnet generated by $X = P_A \cup \{p^0\} \cup T$ is an S^2P.*
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 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. ${}^{\bullet\bullet}(p^0) \cap P_R = (p^0)^{\bullet\bullet} \cap P_R = \emptyset$.

Note that ${}^\bullet r$ represents place r's input transitions. ${}^{\bullet\bullet}r = \cup_{t \in {}^\bullet r} {}^\bullet t$ is the set of all input places of all input transitions of place r. Similarly, $r^{\bullet\bullet} = \cup_{t \in r^\bullet} t^\bullet$ represents the set of all output places of all output transitions of place r. For example, in Fig. 1.8 (d), ${}^\bullet p_7 = \{t_2, t_6\}$ and ${}^{\bullet\bullet}p_7 = {}^\bullet t_2 \cup {}^\bullet t_8 = \{p_1, p_5, p_8\}$. $p_7^\bullet = \{t_1, t_5\}$ and $p_7^{\bullet\bullet} = t_1^\bullet \cup t_7^\bullet = \{p_1, p_5, p_8\}$. Clearly, ${}^{\bullet\bullet}p_7 = p_7^{\bullet\bullet}$.

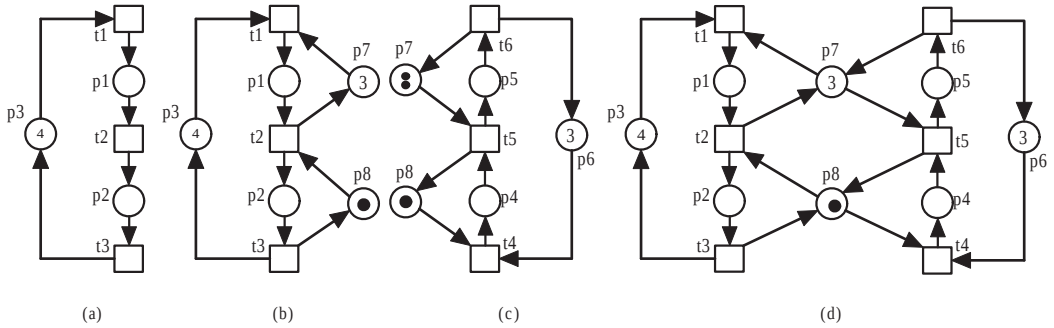


FIG. 1.8 – (a) An S^2P, (b) a marked Petri net (N_1, M_{10}), (c) a marked Petri net (N_2, M_{20}), (d) the composed S^3PR.

Definition 1.19 *[75] Let* $N = (P_A \cup \{p^0\} \cup P_R, T, F)$ *be an* S^2PR. *An initial marking* M_0 *is called an acceptable initial marking for N if (1)* $M_0(p^0) \geq 1$, *(2)* $M_0(p) = 0$, $\forall p \in P_A$, *and (3)* $M_0(r) \geq 1$, $\forall r \in P_R$. *An* S^2PR *with such a marking is said to be an acceptably marked one.*

Exemple 1.7 *The Petri net shown in Fig. 1.8 (a) is an* S^2P, *where* p_1 *and* p_2 *are activity places and* p_3 *is a idle process place. Clearly, the net is a strongly connected state machine.*

The net (N_1, M_{10}) *depicted in Fig. 1.8 (b) is an* S^2PR *extending from the* S^2P *in Fig. 1.8(a), where* $P_R = \{p_7, p_8\}$ *is the set of resource places. It meets the conditions in Definitions 1.18 and 1.19. Hence,* (N_1, M_{10}) *is an* S^2PR *with an acceptable initial marking.*

Definition 1.20 *[75] A system of* S^2PR, *called* S^3PR *for short, is defined recursively as follows :*

1. *An* S^2PR *is an* S^3PR.

2. *Let* $N_i = (P_{A_i} \cup \{p_i^0\} \cup P_{R_i}, T_i, F_i)$, $i \in \{1, 2\}$, *be two* S^3PR *such that* $(P_{A_1} \cup \{p_1^0\}) \cap (P_{A_2} \cup \{p_2^0\}) = \emptyset$, $P_{R_1} \cap P_{R_2} = P_C \neq \emptyset$, *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 *defined as follows :*

   (a) $P_A = P_{A_1} \cup P_{A_2}$.

   (b) $P^0 = \{p_1^0\} \cup \{p_2^0\}$.

   (c) $P_R = P_{R_1} \cup P_{R_2}$.

   (d) $T = T_1 \cup T_2$.

   (e) $F = F_1 \cup F_2$ *is also an* S^3PR.

In the sequel, an S^3PR N composed by n S^2PR N_1-N_n, denoted by $N = \bigcirc_{i=1}^{n} N_i$, is defined as follows : $N = N_1$ if $n = 1$; $N = (\bigcirc_{i=1}^{n-1} N_i) \circ N_n$ if $n > 1$. $\overline{N}_i$ is used to denote the S^2P from which the S^2PR N_i is formed. Transitions in $(P^0)^\bullet$ are called source transitions that represent the entry of raw materials when a manufacturing system is modeled with an S^3PR.

Definition 1.21 *[75] Let* N *be an* S^3PR. (N, M_0) *is called an acceptably marked* S^3PR *if one of the following statements is true :*

1. (N, M_0) *is an acceptably marked* S^2PR.
2. $N = N_1 \circ N_2$, *where* (N_i, M_{0_i}) *is an acceptably marked* S^3PR *and*

   (a) $\forall i \in \{1, 2\}$, $\forall p \in P_{A_i} \cup \{p_i^0\}$, $M_0(p) = M_{0_i}(p)$.

   (b) $\forall i \in \{1, 2\}$, $\forall r \in P_{R_i} \setminus P_C$, $M_0(r) = M_{0_i}(r)$.

   (c) $\forall r \in P_C$, $M_0(r) = max\{M_{0_1}(r), M_{0_2}(r)\}$.

Exemple 1.8 *The net* (N_1, M_{10}) *shown in Fig. 1.8(b) is an* S^3PR *if* p_3 *is an idle process place,* p_1 *and* p_2 *are activity places, and* p_7 *and* p_8 *are resource places. Likewise,* (N_2, M_{20}) *shown in Fig. 1.8(c) is an* S^3PR *if* p_6 *is an idle process place,* p_4 *and* p_5 *are activity places, and* p_7 *and* p_8 *are resource places. Since they have common resource places, they are composable. Their composition leads to an* S^3PR (N, M_0), *as shown in Fig. 1.8(d). Since one can verify that it meets the conditions in Definition 1.21, the net in Fig. 1.8(d) is an acceptably marked* S^3PR.

In what follows, when we talk about an S^3PR, it is assumed to be acceptably marked unless otherwise stated.

Let S be an SMS in an S^3PR $N = (P_A \cup P^0 \cup P_R, T, F)$. Ezpeleta *et al.* [23] show that S does not contain idle process places but consists of activity and resource places only. As a result, S can be represented by $S_A \cup S_R$, where $S_R = S \cap P_R$ and $S_A = S \setminus S_R$, i.e., $S_A = S \cap P_A$.

Definition 1.22 *[75] For $r \in P_R$, $H(r) = {}^{\bullet\bullet}r \cap P_A$, the activity places that use r, is called the set of holders of r. Let $[S] = (\cup_{r \in S_R} H(r)) \setminus S$. $[S]$ is called the complementary set of siphon S.*

The concept of the complementary set of a siphon plays an important role in the development of the deadlock prevention policy in [23]. Intuitively, the complementary set of a siphon is a set of activity places that use the resources in it but are excluded from it. That is to say, the activity places in the complementary set compete for the limited resources with those in the siphon. When the tokens initially staying in the resource places of a siphon are completely held or "stolen" by the places in its complementary set, the siphon is emptied. As is known, if a siphon has no token, it remains free of tokens in the subsequent reachable markings. The transitions in its postset are completely disabled, leading to deadlocks.

Definition 1.23 *[76] Let $\{r_1, r_2, \cdots, r_m\} \subseteq P_R$ $(m \geq 2)$ be a set of resources in an S^3PR $N = (P_A \cup P^0 \cup P_R, T, F)$. A simple circuit $C(r_1, t_1, r_2, t_2, \cdots, r_m, t_m)$ in N is called a resource circuit if*

1. $\forall i \in \mathbb{N}_m$, $r_i \in {}^{\bullet} t_i$;
2. $\forall i \in \{2, \cdots, m\}$, $r_i \in t_{i-1}^{\bullet}$;
3. $r_1 \in t_m^{\bullet}$.

Theorem 1.3

[72] Let C be a resource circuit in an S^3PR $N = (P_A \cup P_R \cup P^0, T, F)$. $S = C^R \cup \{p | p \in \cup_{r \in C^R} H(r) \wedge (p^{\bullet\bullet} \cap (P_A \cup P^0)) \nsubseteq \cup_{r \in C^R} H(r)\}$ is a siphon in N. Furthermore, if S does not contain the support of any P-semiflow, it is strict minimal.

Exemple 1.9 *As shown in Fig. 1.8(d), (N, M_0) is an S^3PR with $P^0 = \{p_1, p_6\}$, $P_A = \{p_1, p_2, p_4, p_5\}$, and $P_R = \{p_7, p_8\}$. We have $I_{p_7} = p_1 + p_5 + p_7$, $I_{p_8} = p_2 + p_4 + p_8$, $H(p_7) = \{p_1, p_5\}$, and $H(p_8) = \{p_2, p_4\}$. By Definition 1.23, $C(p_7, t_5, p_8, t_2)$ is a resource circuit with $C^R = \{p_7, p_8\}$. $S = \{p_2, p_5, p_7, p_8\}$ is the unique SMS in this net. Its complementary set is $\{p_1, p_4\}$.*

1.6.2 GS^{3}PR Net

Generalized systems of simple sequential processes with resources (GS^{3}PR), a subclass of Petri nets, are defined as below. Note that a GS^{3}PR is a generalized version of an S^{3}PR, that is, the weight of an arc in a GS^{3}PR can be greater than one. A GS^{3}PR becomes an S^{3}PR if the weight of each arc is changed to be one.

Definition 1.24 *[126] A generalized simple sequential process with resources (GS^{2}PR) is a Petri net $N = (P_A \cup \{p^0\} \cup P_R, T, F, W)$ if :*

1. *The subnet generated by $X = P_A \cup \{p^0\} \cup T$ is an S^{2}P ;*
2. *$P_R \neq \emptyset$ ($r \in P_R$ is called a resource or a resource place in a net formalism) and $(P_A \cup \{p^0\}) \cap P_R = \emptyset$;*
3. *$W = W_A \cup W_R$, where $W_A : ((P_A \cup P^0) \times T) \cup (T \times (P_A \cup P^0)) \rightarrow \{0, 1\}$ and $W_R : (P_R \times T) \cup (T \times P_R) \rightarrow \mathbb{N}$, where $P^0 = \{p^0\}$.*
4. *$\forall p \in P_A$, $\forall t \in {}^\bullet p$, $\forall t' \in p^\bullet$, ${}^\bullet t \cap P_R = t'^\bullet \cap P_R = \{r_p\}$ and $W(r_p, t) = W(t', r_p)$. $\forall r \in P_R$, $\exists$ a unique minimal P-semiflow I_r s.t. $\|I_r\| = {}^{\bullet\bullet} r \cap P_A \cup \{r\}$, $\|I_r\| \cap P_R = \{r\}$, $\|I_r\| \cap P^0 = \emptyset$, and $I_r(r) = 1$. Furthermore, $P_A = \cup_{r \in P_R}(\|I_r\| \backslash P_R)$;*
5. *The following statements are satisfied : $\forall r \in P_R$, ${}^\bullet r \cap r^\bullet = \emptyset$ and ${}^{\bullet\bullet} r \cap P_A = r^{\bullet\bullet} \cap P_A \neq \emptyset$;*
6. *${}^{\bullet\bullet}(p^0) \cap P_R = (p^0)^{\bullet\bullet} \cap P_R = \emptyset$.*

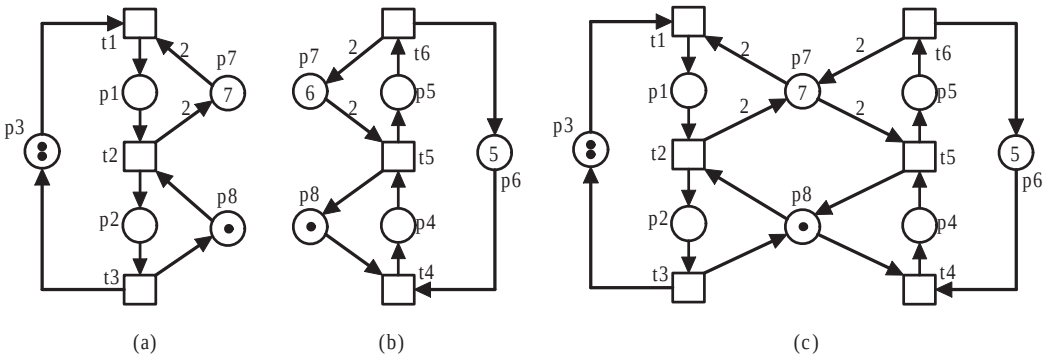


FIG. 1.9 – (a) A marked Petri net (N_1, M_{10}), (b) a marked Petri net (N_2, M_{20}), (c) the composed GS^{3}PR.

The net N_1 shown in Fig. 1.9(a) is a GS^{2}PR extended from the S^{2}P in Fig. 1.8(a), where $P_R = \{p_7, p_8\}$ is the set of resource places. Likewise, N_2 shown in Fig. 1.9(a) is a GS^{2}PR if p_6 is a idle process place, p_4 and p_5 are activity places, and p_7 and p_8 are resource places.

Definition 1.25 *[126] Let $N = (P_A \cup \{p^0\} \cup P_R, T, F, W)$ be a GS^{2}PR. An initial marking M_0 is called an acceptable initial marking for N if : $\forall p \in P_A$, $M_0(p) = 0$; $\forall r \in P_R$, $M_0(r) \geq max_{p \in \|I_r\|} I_r(p)$; and $M_0(p^0) \geq 1$.*

The couple (N, M_0) with N being a GS^{2}PR and M_0 being an acceptable initial marking is called a well-marked GS^{2}PR. An acceptable initial marking guarantees that each transition is potentially fireable. Specially, at least one token in a process idle place means that a process can start. (N_1, M_{10}) depicted in Fig. 1.9(a) meets the conditions in Definitions 1.24 and 1.25. Hence, (N_1, M_{10}) is a GS^{2}PR with an acceptable initial marking.

Definition 1.26 *[126] A system of GS^{2}PR, called GS^{3}PR, is defined recursively as follows :*

1. *A GS^{2}PR is a GS^{3}PR ;*
2. *Let $N_i = (P_{A_i} \cup P_i^0 \cup P_{R_i}, T_i, F_i, W_i)$, $i \in \{1, 2\}$, be two GS^{3}PR such that $(P_{A_1} \cup P_1^0) \cap (P_{A_2} \cup P_2^0) = \emptyset$, $P_{R_1} \cap P_{R_2} = P_C$ $(\neq \emptyset)$, and $T_1 \cap T_2 = \emptyset$ (in this case we say that N_1 and N_2 are two composable GS^{3}PR). Then, the net $N = (P_A \cup P^0 \cup P_R, T, F, W)$ resulting of the composition of N_1 and N_2 via P_C (denoted as $N = N_1 \circ N_2$) defined as follows :*
   - *(a) $P_A = P_{A_1} \cup P_{A_2}$,*
   - *(b) $P^0 = P_1^0 \cup P_2^0$,*
   - *(c) $P_R = P_{R_1} \cup P_{R_2}$,*
   - *(d) $T = T_1 \cup T_2$,*
   - *(e) $F = F_1 \cup F_2$,*
   - *(f) $W = W_1 \cup W_2$ is also a GS^{3}PR.*

Definition 1.27 *[126] Let N be a GS^{3}PR. (N, M_0) is a well-marked GS^{3}PR if one of the two following statements is true :*

1. *(N, M_0) is a well-marked GS^{2}PR ;*

2. $N = N_1 \circ N_2$, *where* (N_i, M_{i0}) *is a well-marked GS*^{2}*PR :*

   (a) $\forall i \in \{1, 2\}$, $\forall p \in P_{A_i} \cup P_i^0$, $M_0(p) = M_{i0}(p)$ *;*

   (b) $\forall i \in \{1, 2\}$, $\forall r \in P_{R_i} \backslash P_C$, $M_0(r) = M_{i0}(r)$ *;*

   (c) $\forall r \in P_C$, $M_0(r) = max\{M_{10}(r), M_{20}(r)\}$.

In the sequel, a GS^{3}PR N composed of k GS^{2}PR $N_1 - N_k$, denoted by $N = \bigcirc_{i=1}^{k} N_i$, is defined as follows : if $k = 1$ then $N = N_1$; if $k > 1$ then $\bigcirc_{i=1}^{k} N_i = (\bigcirc_{i=1}^{k-1} N_i) \circ N_k$, where $k \in \mathbb{N} \backslash \{0\}$. Given N in this way, we denote $\mathcal{I}_N = \{1, ..., k\}$. On the other hand, $\overline{N_i}$ represents the S^{2}P from which we form the GS^{2}PR N_i.

As discussed above, both (N_1, M_{10}) and (N_2, M_{20}) in Fig. 1.9 are GS^{2}PR. Since they have common resource places p_7 and p_8, they are composable. Their composition leads to a GS^{3}PR (N, M_0), as shown in Fig. 1.9(c). Since it meets Definition 1.27, the net in Fig. 1.9(c) is a well-marked GS^{3}PR.

Note that the initial markings in process idle places of GS^{3}PR can affect the liveness of Petri nets. Take the net shown in Fig. 1.9(c) as an example. There are two tokens in process idle place p_3. The net is live. If the initial marking of p_3 is greater than or equal to three, the net is non-live. In this paper, we suppose that an initial marking in a process idle place is greater than or equal to the total capacity of the resources used by the process. Then the process idle place is an implicit place [18], i.e., it can be eliminated without producing any changes in the behavior of the original net.

Definition 1.28 *[126] Let r be a resource place in a GS*^{3}*PR.* $H(r) = {}^{\bullet\bullet} r \cap P_A$*, the activity places that use r, is called the set of holders of r.*

Definition 1.29 *[126] Let S be a siphon in a GS*^{3}*PR with* $S = S_A \cup S_R$, $S_R = S \cap P_R$, *and* $S_A = S \backslash S_R$. $[S] = (\cup_{r \in S_R} H(r)) \backslash S$ *is called the complementary set of S.*

As shown in Fig. 1.9(c), (N, M) is a well-marked GS^{3}PR, where $P^0 = \{p_3, p_6\}$, $P_{A_1} = \{p_1, p_2\}$, $P_{A_2} = \{p_4, p_5\}$, and $P_R = \{p_7, p_8\}$. $S = \{p_2, p_5, p_7, p_8\}$ is its unique SMS, where $S_R = \{p_7, p_8\}$ and $S_A = \{p_2, p_5\}$. We have $I_1 = p_1 + p_2 + p_3$, $I_2 = p_4 + p_5 + p_6$, $I_{p_7} = 2p_1 + 2p_5 + p_7$, $I_{p_8} = p_2 + p_4 + p_8$, $H(p_7) = \{p_1, p_5\}$, and $H(p_8) = \{p_2, p_4\}$. We can obtain $[S] = \{p_1, p_4\}$.

Similar to Proposition IV.2 in [23] (Let $N = (P_A \cup P^0 \cup P_R, T, F)$ be an S^{3}PR and S be a siphon such that it does not contain the support of any P-semiflow. Then, $|S \cap P_R| > 1$), we conclude that a strict siphon in a GS^{3}PR contains at least two resource places.

Theorem 1.4

Let $N = (P_A \cup P^0 \cup P_R, T, F, W)$ be a GS^{3}PR and S be a strict siphon. Then, $|S \cap P_R| \geq 2$.

Proof : A GS^{3}PR N is composed of k GS^{2}PR $N_1 - N_k$, denoted by $N = \bigcirc_{i=1}^{k} N_i$. Net $\overline{N_i}$ represents the S^{2}P from which we form the GS^{2}PR N_i.

Suppose that $S \cap P_R = \emptyset$. $\forall i \neq j \in \mathcal{I}_N, T_i \cap T_j = \emptyset$ is true since each $\overline{N_i}$ is a strongly connected state machine. We can conclude that there exists $i \in \mathcal{I}_N$ such that $(P_i \cup P_i^0) \subseteq S$, and then, S contains the support of a P-semiflow, which is not possible. Then, $|S \cap P_R| > 0$.

Suppose $r \in S \cap P_R$. $H(r) \not\subseteq S$ is true since $H(r) \cup \{r\}$ is the support of a P-semiflow. Let $p \in H(r) \backslash S$. Based on Definition 1.24, let $\{t\} = p^\bullet \cap^\bullet r$. Since S is a siphon, t is necessarily in the postset of the siphon. By $p \notin S$, t is necessarily in the postset of some resource in the siphon. Then, we have ${}^\bullet t \cap P_R = \{r'\} \subset S$. Considering Definition 1.24.5b), i.e., $\forall r \in P_R$, ${}^\bullet r \cap r^\bullet = \emptyset$, we conclude that $r \neq r'$, and then, $\{r, r'\} \subseteq S$, i.e., $|S \cap P_R| \geq 2$. □

Note that the above proof is motivated by the proof in [23]. Based on Theorem 1.4, there necessarily exists the structure in strict siphons of a GS^{3}PR, as shown in Fig. 1.10. Note that this structure is important to understand the proposed new controllability condition in this thesis.

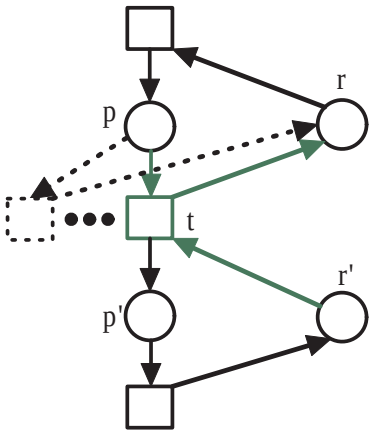


FIG. 1.10 – Typical structure of a siphon in GS^{3}PR (N, M_0).

1.6.3 S^4R Net

This section defines S^4R [81], a class of generalized Petri nets with more powerful modeling ability than S^3PR and GS^3PR. It is equivalent to S^4PR [100] and S^3PGR^2 [89].

Definition 1.30 *[81] A well-marked S^4R net (N, M_0) is a marked Petri net $N = (P, T, F, W)$ with initial marking M_0 such that :*

1. *$P = P_A \cup P^0 \cup P_R$ is a partition such that (1) $P_A = \cup_{i=1}^{n} P_{A_i}$ is called the set of activity places, where $\forall i, j \in \{1, \ldots, n\}$, $i \neq j$, $P_{A_i} \neq \emptyset$ and $P_{A_i} \cap P_{A_j} = \emptyset$. (2) $P^0 = \cup_{i=1}^{n} \{p_i^0\}$ is called the set of idle places. (3) $P_R = \cup_{i=1}^{n} P_{R_i} = \{r_1, r_2, \cdots, r_m\}$ is called the set of resource places ;*
2. *$T = \cup_{j=1}^{n} T_j$, $T_j \neq \emptyset$. $\forall i \neq j$, $T_i \cap T_j = \emptyset$;*
3. *$W = W_A \cup W_R$, where $W_A : ((P_A \cup P^0) \times T) \cup (T \times (P_A \cup P^0)) \rightarrow \{0, 1\}$ and $\forall j \neq i$, $W_A : ((P_{A_j} \cup \{p_j^0\}) \times T_i) \cup (T_i \times (P_{A_j} \cup \{p_j^0\})) \rightarrow \{0\}$, and $W_R : (P_R \times T) \cup (T \times P_R) \rightarrow \mathbb{N}$;*
4. *N_j generated by $P_{A_j} \cup \{p_j^0\} \cup T_j$ is a strongly connected state machine such that every circuit in N_j contains place p_j^0 ;*
5. *$\forall r \in P_R$, there exists a unique minimal P-semiflow I_r such that $\|I_r\| \cap P_R = \{r\}$, $\|I_r\| \cap P^0 = \emptyset$, and $I_r(r) = 1$. Furthermore, we have $P_A = \cup_{r \in P_R} (\|I_r\| \backslash P_R)$;*
6. *N is pure and strongly connected ;*
7. *$\forall p \in P_A$, $M_0(p) = 0$; $\forall r \in P_R$, $M_0(r) \geq max_{p \in \|I_r\|} I_r(p)$; and $\forall p_j^0 \in P^0$, $M_0(p_j^0) \geq 1$.*

Definition 1.31 *[75] Let S be an SMS in an S^4R with $S = S_A \cup S_R$, $S_R = S \cap P_R$, and $S_A = S \backslash S_R$. For $r \in P_R$, $H(r) = \|I_r\| \backslash \{r\}$, the activity places that use r, is called the set of holders of r. $[S] = (\cup_{r \in S_R} H(r)) \backslash S$ is called the complementary set of S.*

As shown in Fig. 1.11, (N, M_0) is a well-marked S^4R with $P^0 = \{p_{10}, p_{11}\}$, $P_A = \{p_1 - p_6\}$, and $P_R = \{r_1, r_2, r_3\}$. The net has three SMS : $S_1 = \{p_3, p_5, p_6, r_2, r_3\}$, $S_2 = \{p_3, p_6, r_1, r_2, r_3\}$, and $S_3 = \{p_2, p_6, r_1\}$. We have $I_{r_1} = p_1 + 3p_2 + p_6 + r_1$, $I_{r_2} = p_2 + 2p_5 + 2p_6 + r_2$, $I_{r_3} = p_3 + p_4 + r_3$, $H(r_1) = \{p_1, p_2, p_6\}$, $H(r_2) = \{p_2, p_5, p_6\}$, and $H(r_3) = \{p_3, p_4\}$.

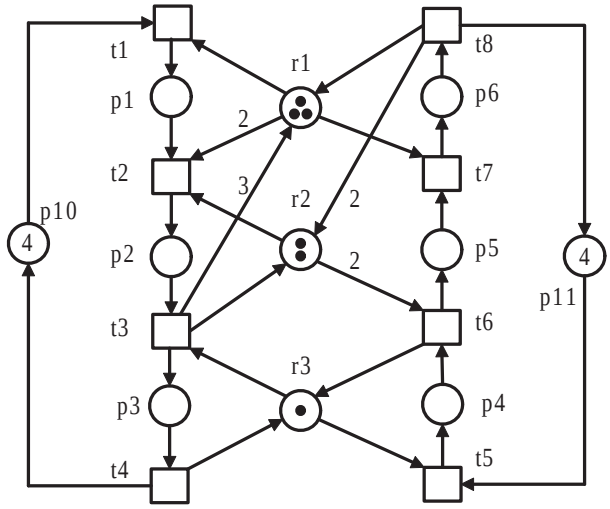


FIG. 1.11 – An S^4R net (N, M_0).

1.6.4 Relationships among S^3PR, GS^3PR, and S^4R Nets

S^3PR and S^4R (S^4R in [81] are equivalent to S^4PR in [100], and S^3PGR^2 in [89]) are typical classes of ordinary Petri nets and generalized Petri nets, respectively. Both S^3PR and S^4R are composed of a set of state machines and a set of resource places. In S^3PR, only one shared resource is allowed to be used at each stage in a job. Compared with the usage of resources in S^3PR, the usage of resources in S^4R is almost arbitrary and requires only conservativeness[1].

GS^3PR is a subclass of S^4R and a generalized version of an S^3PR. A GS^3PR is equivalent to a WS^3PR in [126]. Since any Petri net has a weight function, it is sound and rational to rename a *weighted* S^3PR in [126] to be a *generalized* S^3PR. A GS^3PR becomes an S^3PR if the weight of each arc is changed to be one. Figure 1.7 depicts intuitive examples of the Petri net subclasses mentioned-above. We can see that in an S^3PR, an activity place (representing a processing stage) of a job needs a single unit of a single resource type. For example, the activity modeled with place p_1 in Figure 1.7(a) needs one unit of only one resource type, i.e., p_7. In a GS^3PR, an activity place of a job may need multiple units of a single resource type. For instance, the activity place

[1]The resources in flexible manufacturing systems are *conservative*. A resource has a capacity of processing units that are represented by the tokens in a place that models the resource. A resource is either idle, implying that there is no ongoing processing stage that needs it, or some or all units are occupied by processing stages. In the context of a Petri net model, the tokens initially marked in a resource place are either in the resource place or in the activity places that need the resource to support.

modeled with p_1 in Figure 1.7(b) needs two units of only one resource type, i.e., p_7. While, in an S^4R, an activity place may be supported by any resource requirements, specifically, multiple units of multiple resource types. In Figure 1.7(c), the activity place modeled with p_3 needs one unit of resource p_8 and one unit of resource p_9.

GS^3PR is a subclass of S^4R and a generalized version of an S^3PR. It is easy to understand that the decision conditions for S^4R still hold for GS^3PR.

1.7 Summary

Petri nets are suitable to describe discrete event systems. Their various analysis techniques make it possible to reveal many behavioral properties of such systems. Basic definitions and properties of Petri nets are given in this chapter. Also, three typical classes of Petri nets are outlined. This chapter is fundamental to understand the ideas presented in the following ones.

Chapitre 2

Structural Analysis of GS^{3}PR Nets

2.1 Introduction

Structural analysis is one of the most important and efficient methods to investigate the behavior of Petri nets. Liveness is a significant behavioral property of Petri nets. Siphons, as structural objects of a Petri net, are closely related to its liveness. Many deadlock control policies for AMS modeled by Petri nets are implemented via siphon control. Most of the existing methods design liveness-enforcing supervisors by adding control places for siphons based on their controllability conditions. To compute a liveness-enforcing supervisor with as much as permissive behavior, it is both theoretically and practically significant to find an exact controllability condition for siphons. However, the existing conditions, max, max$'$, and max$''$-controllability of siphons are all overly restrictive and generally sufficient only. This chapter develops a new condition called max*-controllability of the siphons in GS^{3}PR, which are a net subclass that can model many real-world automated manufacturing systems. We show that a GS^{3}PR is live iff all its SMS are max*-controlled. Compared with the existing conditions, i.e., max-, max$'$-, and max$''$-controllability of siphons, max*-controllability of the SMS is not only sufficient but also necessary.

Then, for GS^{3}PR, an integer programming (IP) model is formulated, which can detect the existence of minimal non-max*-marked siphons that cause deadlocks or livelocks. We conclude that a GS^{3}PR is live if there is no feasible solution to the formulated IP model.

The rest of this chapter is organized as follows. Section 2 motivates this study via an example. Section 3 formulates the new concept called max*-controlled siphons and concludes that a GS^{3}PR is live iff all its SMS are max*-controlled. Differences among the max, max$'$, max$''$, and max*-

controllability of siphons are discussed in Section 4. Section 5 develops a general IP test for liveness detection for GS^{3}PR based on max*-controllability condition. Examples are used to illustrate the proposed method in Section 6. Also, we further discuss the proposed IP in this section. Finally, Section 7 concludes this chapter.

2.2 Motivation

In this section, we first briefly review the concepts of the max-, max $'$-, and max$''$-controllability conditions of a siphon in a GS^{3}PR net in [4], [9], [126], and [80] by an example, which motivates this study.

2.2.1 Max-controlled Siphons

The following properties of GS^{3}PR nets are from [4]. Given a place p, we denote $max_{t\in p^\bullet}\{W(p, t)\}$ by $max_{p^\bullet}$.

Definition 2.1 *[4] Let S be a siphon of a net system (N, M_0). S is said to be max-marked at $M \in R(N, M_0)$ if $\exists p \in S$ such that $M(p) \geq max_{p^\bullet}$.*

Definition 2.2 *[4] Let S be a siphon in a well-marked net (N, M_0). S is said to be max-controlled if S is max-marked at any reachable marking $M \in R(N, M_0)$.*

Definition 2.3 *[4] (N, M_0) satisfies the max cs-property (controlled siphon-property) if each minimal siphon of N is max-controlled.*

Theorem 2.1

[4] A generalised Petri net is deadlock-free if it satisfies the max cs-property.

Corollary 2.1

A GS^{3}PR net is deadlock-free if it satisfies the max cs-property.

Proof : It follows from Theorem 2.1 by considering that a GS^{3}PR net is a generalised Petri net. □

As shown in Fig. 2.1(a), (N, M_0) is a well-marked GS^{3}PR, where $P^0 = \{p_{11}, p_{12}, p_{13}, p_{14}\}$, $P_A = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8\}$, and $P_R = \{p_9, p_{10}\}$. $S = \{p_2, p_4, p_6, p_8, p_9, p_{10}\}$ is its unique

SMS, where $S_R = \{p_9, p_{10}\}$ and $S_A = \{p_2, p_4, p_6, p_8\}$. In this net, $M_0(p_9) = 9$ and $max_{p_9^\bullet} = max\{W(p_9, t_1), W(p_9, t_4), W(p_9, t_8), W(p_9, t_{11})\} = max\{5, 4, 1, 6\} = 6$. We have $M_0(p_9) > max_{p_9^\bullet}$. Hence, S is max-marked at M_0. Suppose that both t_1 and t_{10} fire once. Then, the net system reaches M with $M = p_1 + p_7 + 4p_9 + 4p_{10} + 2p_{11} + 11p_{12} + 10p_{13} + p_{14}$ as shown in Fig. 2.1(b). We have $M(p_9) = 4 < max_{p_9^\bullet} = 6$ and $M(p_{10}) = 4 < max_{p_{10}^\bullet} = 6$. Based on the definition of the max-marked siphons, S is not max-marked at this marking. Therefore, (N, M_0) does not satisfy the max cs-property. However, the net is live by, as can be obtained by, reachability graph analysis. The max-controllability condition for siphons is restrictive in the sense of deadlock control.

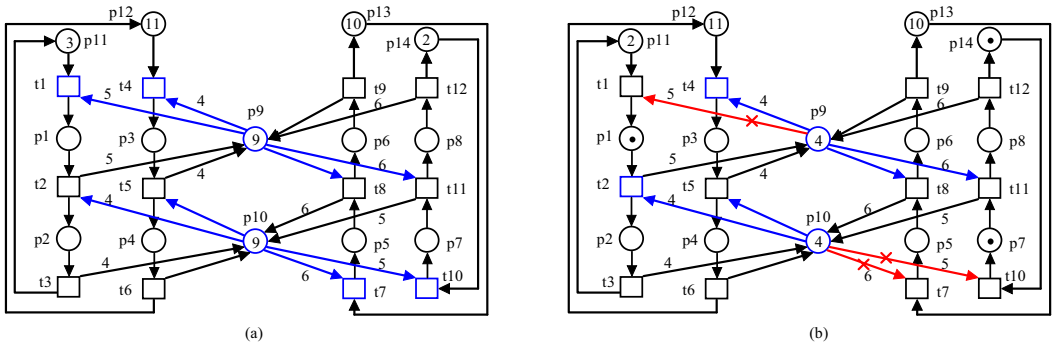


FIG. 2.1 – (a) A GS^{3}PR net (N, M_0), (b) a live GS^{3}PR with a non-max-controlled siphon.

2.2.2 Max′-controlled Siphons

In [9], Chao first proposes a new concept namely max′-controlled siphons to relax the max-controlled condition. [126] refine the concept and propose the formal definition for max′-controlled siphons.

Definition 2.4 *[126] Let S be a siphon in a well-marked S^4R (N, M_0). S is said to be max′-marked at $M \in R(N, M_0)$ if $\exists p \in S_A$ such that $M(p) \geq 1$ or $\exists p \in S_R$ such that $M(p) \geq max_{t \in p^\bullet \cap [S]^\bullet}\{W(p, t)\}$.*

Definition 2.5 *[126] Let S be a siphon in a well-marked S^4R (N, M_0). S is said to be max′-controlled if S is max′-marked at M, $\forall M \in R(N, M_0)$.*

Theorem 2.2

[9] Let (N, M_0) be a well-marked S^4R. If every siphon in the net is max′-controlled, it is live.

Corollary 2.2

Let (N, M_0) be a well-marked GS^{3}PR. If every siphon in the net is max$'$-controlled, it is live.

Proof : It follows from Theorem 2.2 by considering that a GS^{3}PR net is a subclass of S^{4}R. □

The net at $M' = p_1 + p_5 + 4p_9 + 3p_{10} + 2p_{11} + 10p_{12} + 9p_{13} + 2p_{14}$ shown in Fig. 2.2 is obtained by firing t_1 and t_7 once in Fig. 2.1(a). $M'(p_9) = 4$, $M'(p_{10}) = 3$, $max_{p_9^\bullet \cap [S]^\bullet} = max\{W(p_9, t_8), W(p_9, t_{11})\} = max\{1, 6\} = 6$, and $max_{p_{10}^\bullet \cap [S]^\bullet} = max\{W(p_{10}, t_2), W(p_{10}, t_5)\} = max\{4, 1\} = 4$. We have $M'(p_9) < max_{p_9^\bullet \cap [S]^\bullet}$ and $M'(p_{10}) < max_{p_{10}^\bullet \cap [S]^\bullet}$. Based on the definition of the max$'$-marked siphons, S is not max$'$-marked at this marking. However, the net is live. The max$'$-controllability condition for siphons is still restrictive in the sense of deadlock control.

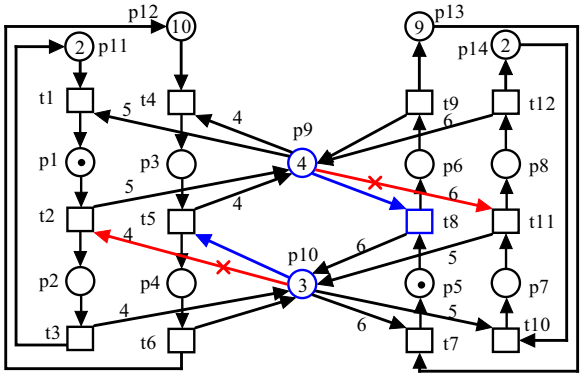


FIG. 2.2 – A live GS^{3}PR with a non-max$'$-controlled siphon.

2.2.3 Max$''$-controlled Siphons

In our previous work [80], we presented a new concept called max$''$-controllability condition of siphons to relax the max$'$-controllability condition.

Definition 2.6 *[80] Let S be a siphon in a well-marked S^{4}R (N, M_0). S is said to be max$''$-marked at $M \in R(N, M_0)$ if at least one of the following conditions holds :*

(i) M is an initial marking ;

(ii) $\exists p \in S_A$ such that $M(p) \geq 1$;

(iii) $\exists r \in S_R$, $min \sum_{t\in T'} \alpha_t \cdot W(t,r) + M(r) \geq max_{t'\in r^\bullet \cap [S]^\bullet}\{W(r,t')\}$, *where* $T' = \{t | t \in {}^\bullet r \cap [S]^\bullet,\ \forall r' \in {}^\bullet t \cap P_R, M(r') \geq W(r',t),\ M(P_A \cap {}^\bullet t) \geq 1\}$, α_t *denotes the times that t is fired from marking M, and* $min \sum_{t\in T'} \alpha_t \cdot W(t,r)$ *can be solved by the following mixed integer program (MIP) :*

$$min \sum_{t\in T'} \alpha_t \cdot W(t,r)$$

$p \in {}^\bullet t \cap P_A$, $M(p) \geq 1$, $t_x \in p^\bullet \cap T'$

$$\sum \alpha_{t_x} \leq M(p)$$

$r' \in {}^\bullet t \cap P_R$, $t_y \in r'^\bullet \cap T'$

$$\sum \alpha_{t_y} \cdot W(r', t_y) \leq M(r')$$

$t \in {}^\bullet r \cap [S]^\bullet$

$$min\{\frac{M(r') - \sum \alpha_{t_y} \cdot W(r',t_y)}{W(r',t)}, M(p) - \sum \alpha_{t_x}\} < 1$$

$$\alpha_t \in \mathbb{N}$$

Definition 2.7 *[80] Let S be a siphon in a well-marked* S^4R (N, M_0). *S is said to be max″-controlled if* $\forall M \in R(N, M_0)$, *S is max″-marked at M.*

Theorem 2.3

[80] Let (N, M_0) *be a well-marked* S^4R. *The net is live if all its siphons are max″-controlled.*

Corollary 2.3

Let (N, M_0) *be a well-marked* GS^3PR. *The net is live if all its siphons are max″-controlled.*

Proof : It follows from Theorem 2.3 by considering that a GS^3PR net is a subclass of S^4R. □

The net at marking $M'' = p_1 + p_3 + p_5 + 3p_{10} + 2p_{11} + 10p_{12} + 9p_{13} + 2p_{14}$ shown in Fig. 2.3 is obtained by firing t_1, t_4, and t_7 once in Fig. 2.1(a). $M''(p_9) = 0$, $M''(p_{10}) = 3$, $M''(p_3) = 1$, and $T' = \{t_5\}$. We have $min\alpha_{t_5} \cdot W(t_5, p_9) + M''(p_9) = 4\alpha_{t_5} = 4 < max_{p_9^\bullet \cap [S]^\bullet}$, where α_{t_5} can be obtained by solving the MIP in Definition 2.6. Based on the definition of the max″-marked siphons, this S is not max″-marked at this marking. However, the net is live. The max″-controllability condition for siphons is still restrictive.

By Definitions 2.1, 2.4, and 2.6, we check siphon $\{p_2, p_4, p_6, p_8, p_9, p_{10}\}$ at markings shown in Fig.s 2.1(a), 2.1(b), 2.2, and 2.3. Test results are shown in Table 2.1. We can see that the above

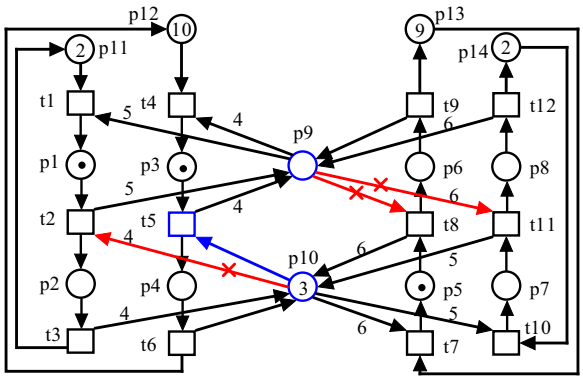


FIG. 2.3 – A live GS^{3}PR with a non-max $''$-controlled siphon.

three controllability conditions of siphons are all sufficient but not necessary for the liveness of GS^{3}PR. This study aims to relax the controllability condition by proposing the max*-controlled siphons in GS^{3}PR.

TAB. 2.1 – Controllability of siphon $\{p_2, p_4, p_6, p_8, p_9, p_{10}\}$ at different markings

the marking in	max-marked	max$'$-marked	max$''$-marked
Fig. 2.1(a)	yes	yes	yes
Fig. 2.1(b)	no	yes	yes
Fig. 2.2	no	no	yes
Fig. 2.3	no	no	no

2.3 Necessary and Sufficient Condition

In order to facilitate understanding and readability, we first define critical transitions in a GS^{3}PR net. Then the definition of a max*-controlled siphons is introduced. Finally, we present the main results.

Definition 2.8 *Let S be a strict siphon in a well-marked GS^{3}PR net (N, M_0) and $[S]$ its complementary set. $T_r^c = r^\bullet \cap [S]^\bullet$ is called the set of critical transitions of r, where $r \in S_R$. $T_S^c = S_R^\bullet \cap [S]^\bullet$ is called the set of critical transitions of S.*

Theorem 2.4

Let S be a strict siphon in a well-marked GS^{3}PR net (N, M_0) and $[S]$ its complementary set. $T_S^c = S_R^\bullet \cap [S]^\bullet = [S]^\bullet$.

Proof : Let $r \in S_R$ and $p \in H(r) \backslash S$, i.e., $p \in [S]$. From Definition 1.24, for each $t \in [S]^\bullet$, we can find $\{t\} = p^\bullet \cap^\bullet r$. Since S is a siphon, t is necessarily in the postset of the siphon. By $p \notin S$, t is necessarily in the postset of some resource in the siphon. Then we have ${}^\bullet t \cap P_R = \{r'\} \subset S$. Hence, $t \in S^\bullet$. In a word, for each $t \in [S]^\bullet$, we can find $t \in S_R^\bullet$, i.e., $T_S^c = S_R^\bullet \cap [S]^\bullet = [S]^\bullet$. □

$t \in T_S^c$ is a critical transition of S. The relationship between T_S^c and T_r^c can be written as $T_S^c = \bigcup_{r \in S_R} T_r^c$. From Theorem 1.4 and Fig. 1.10, the fact that a transition $t \in T_S^c$ is enabled means $e_{pt} = 1$ and $e_{rt} = 1$, where $p \in [S] \cap^\bullet t$ and $r \in S \cap^\bullet t$. The GS^{3}PR net shown in Fig. 2.1(a) is well-marked. $S = \{p_2, p_4, p_6, p_8, p_9, p_{10}\}$ is its unique SMS with $[S] = \{p_1, p_3, p_5, p_7\}$. $T_{p_9}^c = p_9^\bullet \cap [S]^\bullet = \{t_1, t_4, t_8, t_{11}\} \cap \{t_2, t_5, t_8, t_{11}\} = \{t_8, t_{11}\}$, $T_{p_{10}}^c = p_{10}^\bullet \cap [S]^\bullet = \{t_2, t_5, t_7, t_{10}\} \cap \{t_2, t_5, t_8, t_{11}\} = \{t_2, t_5\}$, and $T_S^c = T_{p_9}^c \cup T_{p_{10}}^c = \{t_2, t_5, t_8, t_{11}\}$.

Definition 2.9 *Let S be a strict siphon in a well-marked GS^{3}PR net (N, M_0). S is said to be max*-marked (non-max*-marked) at $M \in R(N, M_0)$ if at least one (none) of the following conditions holds :*

(i) $\exists p \in S_A$, $M(p) \geq 1$;

(ii) $\exists r \in S_R$, $M(r) \geq max_{t \in T_r^c} W(r, t)$;

(iii) $\exists t \in T_S^c$, $e_{pt} = 1$ *and* $e_{rt} = 1$ *(t is enabled at M).*

This definition presents a new concept called a max*-marked siphon. In essence, Conditions (i) and (ii) are completely identical with the conditions in Definition 2.4. It is easy to find that the proposed definition admits an extra condition. Understandably, it is more general than the definition of max$'$-marked siphons. Based on the definition of GS^{3}PR nets and Theorem 1.4, there necessarily exists the typical structure of a siphon in a GS^{3}PR net as shown in Fig. 1.10. In this structure, $e_{pt} = 1$ and $e_{rt} = 1$ in Condition (iii) mean that t is enabled at M. At marking M, a max*-marked siphon can guarantee that at least one transition in its preset can potentially fire once. The GS^{3}PR net shown in Fig. 2.1(a) is well-marked. By firing t_1, t_4, and t_7 once, we can obtain M as shown in Fig. 2.3 with $M = p_1 + p_3 + p_5 + 3p_{10} + 2p_{11} + 10p_{12} + 9p_{13} + 2p_{14}$. $p_3 \in {}^\bullet t_5 \cap P_A$, $M(p_3) = 1$, and $e_{p_3 t_5} = 1$. $\{p_{10}\} =^\bullet t_5 \cap P_R$, $M(p_{10}) = 3$, and $e_{p_{10} t_5} = 1$. $t_5 \in T_S^c$, $e_{p_3 t_5} = 1$, and $e_{p_{10} t_5} = 1$ satisfy the third condition in Definition 2.9. Hence, S is max*-marked at M.

Theorem 2.5

Let S be a strict siphon in a well-marked GS^{3}PR net (N, M_0), which is non-max*-marked at $M \in$

$R(N, M_0)$. The following statements are true :

(1) M is a dead marking with respect to S if $\forall t \in S^\bullet \backslash {}^\bullet S$, t is disabled at M.

(2) $\forall t \in T'_S$, t is dead at M, where $T'_S = \{t | t \in T^c_S, r' \in {}^\bullet t \cap S_R, M(r') < W(r', t), p \in {}^\bullet t \cap P_A, M(p) \geq 1\}$.

(3) $\forall r \in S_R$, $M(r) \geq M'(r)$, where $M' \in R(N, M)$.

Proof : (1) As known, S is non-max*-marked at $M \in R(N, M_0)$. From Definition 2.9, the following statements hold : (i) $\forall p \in S_A$, $M(p) = 0$; (ii) $\forall r \in S_R$, $M(r) < max_{t \in T^c_r} W(r, t)$; and (iii) $\forall t \in T^c_S$, t is disabled at M. This means that only $t \in S^\bullet \backslash {}^\bullet S$ is possibly enabled. By the assumption that $\forall t \in S^\bullet \backslash {}^\bullet S$, t is disabled at M, the fact that M is a dead marking with respect to S can be concluded.

(2) One can obtain $\sum_{r \in S_R} M^T \cdot Y_r = \sum_{p \in [S], r \in S_R} M(p) \cdot Y_r(p) + \sum_{p \in S_A} M(p) \cdot Y_r(p) + M(S_R)$. At marking M, $\forall p \in S_A$, $M(p) = 0$. Hence, we can simplify this equation as $\sum_{r \in S_R} M^T \cdot Y_r = \sum_{p \in [S], r \in S_R} M(p) \cdot Y_r(p) + M(S_R)$, where Y_r is the P-semiflow associated with resource r.

Suppose that $\exists r \in S_R$, $M(r) < M'(r)$, where $M' \in R(N, M)$. Considering the above equation, this assumption means that from marking M to M', some tokens in the complementary set can return to resource place r. This implies that there exists transition t in the preset of r which is enabled at some reachable marking from M. Specifically, $t \in T'_{r'}$ and $T'_{r'} = \{t | t \in {}^\bullet r \cap T^c_{r'}, r' \in {}^\bullet t \cap S_R, M(r') < W(r', t), p \in {}^\bullet t \cap P_A, M(p) \geq 1\}$. In this situation, firing t requires $M''(r') \geq W(r', t)$, where $M'' \in R(N, M)$. The increment of token count in r' requires that some tokens in the complementary set can return to it. This implies that there exists transition t' in the preset of r' which is enabled at some reachable marking from M. Specifically, $t' \in T'_{r''}$ and $T'_{r''} = \{t' | t' \in {}^\bullet r' \cap T^c_{r''}, r'' \in {}^\bullet t' \cap S_R, M(r'') < W(r'', t'), p' \in {}^\bullet t' \cap P_A, M(p') \geq 1\}$. In this situation, firing t' requires $M'''(r'') \geq W(r'', t')$, where $M''' \in R(N, M)$. However, the increment of token count in r'' requires that some tokens in the complementary set can return to it. The number of resource places in S is finite. By Definition 1.24 and Theorem 1.4, the case represented above forms a circular wait[1]. At marking M, no transition in T'_S is enabled, and the same holds for all markings reachable

[1]The concept of circular waits in Petri nets are presented in [52], [68], and [79]. For any two $r_i, r_j \in P_R$, r_i is said to wait for r_j, denoted as $r_i \rightarrow r_j$, if the availability of r_j is an immediate requirement for the release of r_i, or equivalently, if $\exists t \in {}^\bullet r_i \cap r_j^\bullet$. An R-path between r_i and r_k is defined as a set of resource places such that $r_i \rightarrow r_j \rightarrow \cdots \rightarrow r_k$. Then r_i is said to wait over an R-path for r_k, denoted as $r_i \hookrightarrow r_k$, if there is an R-path between r_i and r_k [79]. A circular wait is a set of resource places $C \subseteq P_R$, with $|C| > 1$, such that for any ordered pair $\{r_i, r_j\} \subseteq C$, $r_i \hookrightarrow r_j$ [68].

from M, where $T'_S = \{t | t \in T^c_S, r' \in {}^\bullet t \cap S_R, M(r') < W(r', t), p \in {}^\bullet t \cap P_A, M(p) \geq 1\}$.

(3) By (2), $\forall r \in S_R$, $M(r) \geq M'(r)$, where $M' \in R(N, M)$. □

The theorem reveals that if S is non-max*-marked at $M \in R(N, M_0)$, the tokens flowed in $[S]$ will not be returned to the siphon owing to the insufficient marking of resources at M. The GS^{3}PR net N shown in Fig. 2.4(a) is well-marked at M_0. There are three SMS in this net : $S_1 = \{p_5, p_9, p_{18}, p_{12}, p_{13}\}$, $S_2 = \{p_6, p_4, p_{17}, p_{13}, p_{14}\}$, and $S_3 = \{p_6, p_9, p_{18}, p_{12}, p_{13}, p_{14}\}$. At marking $M = 6p_1 + p_2 + 2p_3 + p_7 + p_8 + 9p_{10} + p_{12} + p_{13} + p_{17} + 9p_{19}$ shown in Fig. 2.4(b), S_1 and S_3 are non-max*-marked siphons, and S_2 is max*-marked siphon.

Take S_1 as an example. We have $[S_1] = \{p_3, p_4, p_{17}\}$, $S_{1R} = \{p_{12}, p_{13}\}$, $S_{1A} = \{p_5, p_9, p_{18}\}$, and $T^c_{S_1} = S^\bullet_{1R} \cap [S_1]^\bullet = \{t_3, t_{10}, t_{14}\}$. For S_1, three conditions in Definition 2.9 are considered. Condition (i) is not satisfied due to $M(p_5) = M(p_9) = M(p_{18}) = 0$. Since $M(p_{12}) = 1 < 2$ ($max_{t \in T^c_{p_{12}}} W(p_{12}, t) = max\{W(p_{12}, t_{10}), W(p_{12}, t_{14})\} = max\{1, 2\} = 2$) and $M(p_{13}) = 1 < 2$ ($max_{t \in T^c_{p_{13}}} W(p_{13}, t) = max\{W(p_{13}, t_3)\} = 2$), Condition (ii) does not hold. Also, Condition (iii) is not satisfied since t_3, t_{10}, and t_{14} are all disabled at M. Now, the three statements in Theorem 2.5 are considered. $t_2 \in S^\bullet_1 \backslash {}^\bullet S_1$ is disabled at M. Hence, M is not a dead marking with respect to S_1. We have $T'_{S_1} = \{t_3, t_{14}\}$. From Fig. 2.4(b), t_3 can fire at M' if tokens in complementary place p_{17} can return to resource place p_{13}, where $M' \in R(N, M)$. This demands that t_{14} should be potentially enabled at $M''(M'' \in R(N, M))$. Firing t_{14} requires $M''(p_{12}) \geq W(p_{12}, t_{14})$. The increment of token count in p_{12} requires that some tokens in complementary place p_3 can return to it. This requires that t_3 should be potentially enabled. The case stated above forms a circular wait $\{p_{12}, p_{13}\}$. Hence, t_3 and t_{14} are disabled at M and they cannot fire again at any $M' \in R(N, M)$. For p_{12} and p_{13}, $M(p_{12}) \geq M'(p_{12})$ and $M(p_{13}) \geq M'(p_{13})$, where $M' \in R(N, M)$.

Theorem 2.6

Let S be a strict siphon in a well-marked GS^{3}PR net (N, M_0). If S is non-max*-marked at a reachable marking, (N, M_0) is non-live.

Proof : Suppose that S is non-max*-marked at reachable marking M. By Theorem 2.5, $\forall t \in T'_S$, t is dead at M, where $T'_S = \{t | t \in T^c_S, r' \in {}^\bullet t \cap S_R, M(r') < W(r', t), p \in {}^\bullet t \cap P_A, M(p) \geq 1\}$. It is clear that $T'_S \neq \emptyset$. That is to say, there exists at least a transition which is dead at M if S is non-max*-marked at this marking. By the definition of a non-live net, the result holds. □

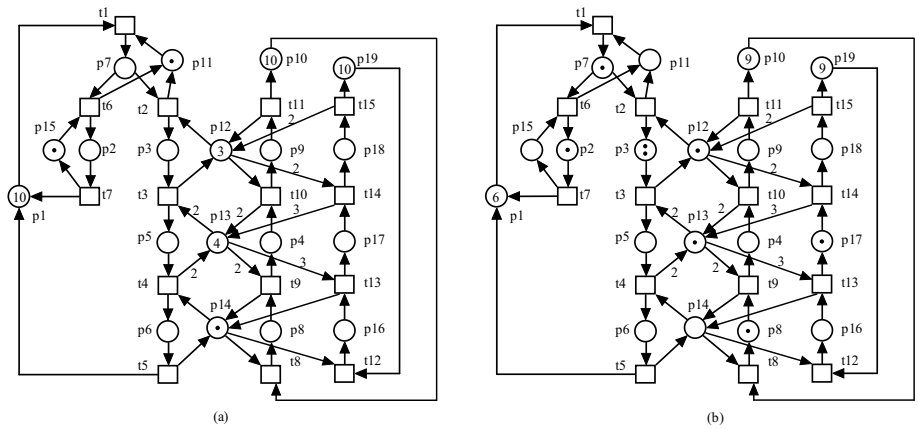


FIG. 2.4 – (a) A GS^{3}PR (N, M_0), (b) the net at $M = 6p_1 + p_2 + 2p_3 + p_7 + p_8 + 9p_{10} + p_{12} + p_{13} + p_{17} + 9p_{19}$.

As shown in Fig. 2.4(b), S_1 is non-max*-marked at the current marking. By Theorem 2.6, the net is non-live.

Definition 2.10 *Let S be a strict siphon in a well-marked GS^{3}PR net (N, M_0). S is said to be max*-controlled if S is max*-marked at any reachable marking from M_0.*

Lemma 2.1

[9] Let (N, M_0) be a well-marked GS^{3}PR net, $M \in R(N, M_0)$ and $t \in T$ a dead transition at M. Then, there exists $M' \in R(N, M)$ and two subsets $\mathcal{J} \subset \mathcal{I}_N$ and $\mathcal{H} \subset \mathcal{I}_N$ such that $\mathcal{I}_N = \mathcal{J} \cup \mathcal{H}$, $\mathcal{I}_N = \{1, 2, ..., n\}$, $\mathcal{J} \cap \mathcal{H} = \emptyset$, $\mathcal{J} \neq \emptyset$ and (i) $\forall h \in \mathcal{H}$, $M'(p_h^0) = M_0(p_h^0)$, (ii) $\forall j \in \mathcal{J}$, $M'(p_j^0) < M_0(p_j^0)$ and $\Omega = \{p^\bullet | p \in P_A, and \ \ M'(p) > 0\}$ is a set of dead transitions.

From Definition 1.26, a GS^{3}PR contains n subnets that are strongly connected state machines such that every circuit contains an idle place. According to Lemma 2.1, at M', n subnets are divided into two parts. The first cannot proceed to complete operations, i.e., some tokens in their idle places at the initial marking cannot return to them. The second can proceed to complete operations. The former subnets necessarily exist due to the existence of the dead transitions.

Theorem 2.7

Let (N, M_0) be a well-marked GS^{3}PR and $t \in T$ be a dead transition at $M \in R(N, M_0)$. Then there exists an SMS S and a marking $M' \in R(N, M)$ such that S is non-max*-marked at M'.

Proof : Five steps are presented to prove the result.

Step1 : There exists a non-empty places set S^0 at M'.

Consider M' given in Lemma 2.1. Let $S_R^0 = \{r \in P_R | \exists t \in r^\bullet, M'(r) < W(r,t), p \in P_A \cap {}^\bullet t, M'(p) > 0\}$ and $S_A^0 = \{p \in H(r) | r \in S'_R, M'(p) = 0\}$. S_R^0 is the set of resource places with at least one disabled output transition t that is not disabled by its input activity place p. S_A^0 is the set of unmarked holders of these resources. We need to prove that $S' = S_A^0 \cup S_R^0$ is non-empty.

$S^0 \neq \emptyset$. By contradiction, suppose that $S^0 = \emptyset$. Then $S_R^0 = \emptyset$. This implies that $\forall p \in P_A, t \in p^\bullet$ can fire if $M'(p) > 0$. Then the set of indexes J given in Lemma 2.1 is empty. This contradicts Lemma 2.1. Hence, $S^0 \neq \emptyset$.

Step2 : S^0 is a strict siphon.

Step2.1 : S^0 is a siphon. Construct a non-empty set S^0 by using dead transitions. Let $t \in {}^\bullet S^0$. Two cases are considered.

case-1 : $t \in {}^\bullet r$ for $r \in S_R^0$: ${}^\bullet t \cap P_A$ contains a unique place since each subnet N_i in GS^{3}PR is a state machine. Let ${}^\bullet t \cap P_A = \{p\}$. If $M'(p) = 0$, then $p \in S_A^0$, which implies that $t \in p^\bullet \subseteq S_A^{0\bullet} \subseteq S^{0\bullet}$. If $M'(p) \geq 1$, then ${}^\bullet t \cap P_R \neq \emptyset$. By contrary, suppose that ${}^\bullet t \cap P_R = \emptyset$. Then, t is enabled at M' due to $M'(p) \geq 1$. By the definition of GS^{3}PR, $|{}^\bullet t \cap P_R| = 1$. We use r' to denote the unique element in ${}^\bullet t \cap P_R$. Next, we need to prove that at M', $W(r',t) > M'(r')$ is true. By contradiction, if $W(r',t) \leq M'(r')$, then t is enabled at M', which contradicts the assumption that t is dead. In essence, t is dead due to the existence of its insufficiently marked input resource place at M'. Hence, $r' \in S_R^0$, and then, $t \in r'^\bullet \subseteq S_R^{0\bullet} \subseteq S^{0\bullet}$.

case-2 : $t \in {}^\bullet p$ for $p \in S_A^0$: Since $p \in S_A^0$, $\exists r \in S_R^0$ such that $p \in H(r)$, and $t \in r^\bullet \subseteq S^{0\bullet}$.

Step2.2 : S^0 is a strict siphon.

By contradiction, suppose that S^0 is not strict. There necessarily exists a P-semiflow $\|I_{r''}\| \subseteq S^0$, where $r'' \in S^0$. By the construction of S^0, $M'(r'') < M_0(r'')$ holds. This implies that $\exists q \in \|I_{r''}\| \cap P_A$ such that $M'(q) \geq 1$. Place q cannot be in S^0 since $S_A^0 = \{p \in H(r) | r \in S'_R, M'(p) = 0\}$. Hence, S^0 is a strict siphon.

Step3 : Extract place sets S^m from S^0, where $S^m = S_R^m \cup S_A^m$ with $S_R^m = \{r \in S_R^{m-1} | \exists t \in r^\bullet \cap [S^{m-1}]^\bullet, M'(r) < W(r,t), p \in S_A^{m-1} \cap {}^\bullet t, M'(p) > 0\}$ and $S_A^m = \{p \in H(r) | r \in S_R^m, M'(p) = 0\}$, $[S^k] = [S^{k-1}]$, and $m \in \{1, \cdots, k\}$. k is finite since S^0 is finite. Similar to *Step2*, it is easy to prove that S^k is a strict siphon.

Step4 : S^k is non-max*-marked at M.

Now let us prove that S^k is non-max*-marked at M'. The three conditions in Definition 2.9 are considered here.

(i) $\forall p \in S_A^k, M'(p) = 0$;

(ii) $\forall r \in S_R^k, M'(r) < max_{t \in T_r^c} W(r,t)$;

(iii) By Lemma 2.1, at M', $\Omega = \{p^\bullet | p \in P_A, and \quad M'(p) > 0\}$ is a set of dead transitions. $\forall t \in T_{S^k}^c$, t is disabled at M'. Hence, S^k is non-max$''$-marked.

Step5 : There exists a non-max*-marked SMS such that $S \subseteq S^k$. This results can be obtained directly from *Step4*. □

Here we use the net shown in Fig. 2.4 to illustrate Theorem 2.7. $M' = M = 6p_1 + p_2 + 2p_3 + p_7 + p_8 + 9p_{10} + p_{12} + p_{13} + p_{17} + 9p_{19}$ is a marking satisfying Lemma 2.1. At M', $S_R^0 = \{r \in P_R | \exists t \in r^\bullet, M'(r) < W(r,t), p \in P_A \cap {}^\bullet t, M'(p) > 0\} = \{p_{12}, p_{13}, p_{15}\}$ and $S_A^0 = \{p \in H(r) | r \in S'_R, M'(p) = 0\} = \{p_9, p_{18}, p_4, p_5\}$. Hence, $S^0 = S_R^0 \cup S_A^0 = \{p_{12}, p_{13}, p_{15}, p_9, p_{18}, p_4, p_5\}$. Clearly, it is a strict siphon. $[S^0] = \{p_3, p_{17}\}$ is then derived. According to $S_R^m = \{r \in S_R^{m-1} | \exists t \in r^\bullet \cap [S^{m-1}]^\bullet, M'(r) < W(r,t), p \in S_A^{m-1} \cap {}^\bullet t, M'(p) > 0\}$ and $S_A^m = \{p \in H(r) | r \in S_R^m, M'(p) = 0\}$, $S_R^1 = \{p_{12}, p_{13}\}$ since $p_{15}^\bullet \cap [S^{0\bullet}] = \{t_6\} \cap \{t_3, t_{14}\} = \emptyset$, and $S_A^1 = \{p_9, p_{18}, p_4, p_5\}$. Hence, $S^1 = S_R^1 \cup S_A^1 = \{p_{12}, p_{13}, p_9, p_{18}, p_4, p_5\}$. $[S^1] = \{p_3, p_{17}\}$ is then derived. $k = 1$ since $[S^1] = [S^0]$. Clearly, S^1 is a strict siphon. From the figure, an SMS $S_1 = \{p_5, p_9, p_{18}, p_{12}, p_{13}\} \subseteq S^1$ is non-max*-marked at M'.

Theorem 2.8

Let (N, M_0) be a well-marked GS^{3}PR net and $\Pi \neq \emptyset$ be the set of SMS. It is live iff $\forall S \in \Pi$, S is max*-controlled.

Proof : *(sufficiency) :* If $\forall S \in \Pi$, S is max*-controlled, the net GS^{3}PR (N, M_0) is live.

By contradiction, suppose that the net is non-live. From Theorem 2.7, there exists a marking $M \in R(N, M_0)$ and $t \in T$ such that t is a dead transition at M. Then there exists $M' \in R(N, M)$ and an SMS S such that S is a non-max*-marked siphon at M'. In other words, not all SMS of the net are max*-controlled.

(necessity) : If the net GS^{3}PR (N, M_0) is live, $\forall S \in \Pi$, S is max*-controlled.

By contradiction, suppose that there exists an SMS S and a marking $M \in R(N, M_0)$, at which S is non-max*-marked. From Theorem 2.6, the net is non-live. This contradicts the liveness of the net.

In summary, a well-marked GS^{3}PR net (N, M_0) is live iff each SMS is max*-controlled. □

As stated in Section 2, a GS^{3}PR becomes an S^{3}PR if the weight of each arc is changed to be one. In other words, GS^{3}PR are more general than S^{3}PR that are a subclass of GS^{3}PR. This means that the max*-controllability condition of the siphons in GS^{3}PR can be used in S^{3}PR. The three items in Definition 2.9 for S^{3}PR imply that siphon S is marked. It is easy to verify the fact that an S^{3}PR is live iff all its SMS never become unmarked by Theorem 2.8.

2.4 Comparison

From the definition of max-controlled siphons [4], the number of tokens in each place of a siphon is restricted by the maximal weights of its output arcs, i.e., $M(p) \geq max_{p^\bullet}$. Chao [9] points out that this condition is too restrictive and proposes the concept of max$'$-controlled siphons. As for the marking of resource places considered in a siphon, condition $M(p) \geq max_{p^\bullet}$ is relaxed to be $M(p) \geq max_{t \in p^\bullet \cap [S]^\bullet}\{W(p,t)\}$. Note that max$'$-controlled condition of a siphon is still a sufficient but not necessary.

In our previous work [80], we show that the max$'$-controllability condition can be further relaxed by a new concept called max$''$-controlled siphons. As shown in Definition 2.6, at the marking of the considered resource places of a siphon, condition $M(p) \geq max_{t \in p^\bullet \cap [S]^\bullet}\{W(p,t)\}$ is relaxed to be $min \sum_{t \in T'} \alpha_t \cdot W(t,r) + M(r) \geq max_{t' \in r^\bullet \cap [S]^\bullet}\{W(r,t')\}$. This constraint guarantees that each transition in $r^\bullet \cap [S]^\bullet$ is potentially enabled. However, we need to solve an MIP to decide whether a siphon is max$''$-marked at a marking $M \in R(N, M_0)$. What is more, the max$''$-controllability condition of the siphons in S^{4}R is a sufficient condition only.

In this chapter, we improve our previous work [80] by splitting Condition (iii) of Definition 2.6 into two parts : Condition (ii) and Condition (iii) of Definition 2.9. This method avoids solving MIP problems and loosens the constraint $min \sum_{t \in T'} \alpha_t \cdot W(t,r) + M(r) \geq max_{t' \in r^\bullet \cap [S]^\bullet}\{W(r,t')\}$. It is easy to see that the former two conditions of the definition of max*-marked siphons are completely the same with the definition of max$'$-marked siphons. The new definition allows a new condition.

Understandably, it is more general than max′-marked siphons. Condition (iii) in Definition 2.9 means that a critical transition t of S is enabled at M. At M, a max*-marked siphon can guarantee that at least one transition in its preset can fire once.

The net shown in Fig. 2.1(a) is live with 654 legal markings. S is non-max-marked at 8 markings, non-max′-marked at 4 markings, and non-max′′-marked at one marking as shown in Table 2.2, where the numbers of tokens in p_{11}, p_{12}, p_{13}, and p_{14} are not shown. However, S is max*-marked at all 654 markings. In a word, from a max-controlled siphon to max*-controlled one, constraints for siphon control become more and more weaker.

TAB. 2.2 – Controllability conditions of siphon $\{p_2, p_4, p_6, p_8, p_9, p_{10}\}$ in Fig. 2.1(a)

markings of $p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}$	max	max′	max′′	max*
1,0,1,0,1,0,0,0,0,3	no	no	no	yes
1,0,0,0,1,0,0,0,4,3	no	no	yes	yes
0,0,2,0,1,0,0,0,1,3	no	no	yes	yes
0,0,1,0,1,0,0,0,5,3	no	no	yes	yes
1,0,1,0,0,0,1,0,0,4	no	yes	yes	yes
0,0,1,0,0,0,1,0,5,4	no	yes	yes	yes
0,0,2,0,0,0,1,0,1,4	no	yes	yes	yes
1,0,0,0,0,0,1,0,4,4	no	yes	yes	yes

Based on the max*-controllability condition, we can design supervisors with more permissive behavior for GS^{3}PR nets in theory. Hopefully, this condition can be used in an appropriate policy that leads a supervisor of a Petri net system to be optimal. Then the corresponding manufacturing system is of more flexibility. In this sense, the new concept proposed in this chapter can promote the development of optimal or suboptimal deadlock control. In other words, this necessary and sufficient liveness condition for GS^{3}PR will be considered as an important progress in deadlock control of generalized Petri nets. Due to the complex structures of S^{4}R, finding a necessary and sufficient siphon control condition for S^{4}R is a challenging problem and remains open, which, in our own opinion, still needs many efforts.

2.5 Liveness Detection for GS^{3}PR

Based on Theorem 2.8, an immediate implication is that a minimal non-max*-marked siphon at a marking $M \in R(N, M_0)$ can be determined by the following integer programming problem if

the net is non-live.

Theorem 2.9

Let (N, M_0) be a well-marked GS^{3}PR net system. A minimal non-max*-marked siphon S and a corresponding marking $M \in R_S(N, M_0)$ can be obtained through the following IP formulation :

$$min \sum_{p \in P \setminus P^0} s_p \tag{2.1}$$

subject to

For all $t \in T$, $p \in P$:

$$|t^\bullet| \sum_{p \in {}^\bullet t} s_p \geq \sum_{p \in t^\bullet} s_p \tag{2.2}$$

$$\sum_{p \in P^0} s_p = 0 \tag{2.3}$$

$$\sum_{p \in P_A} s_p \geq 1 \tag{2.4}$$

$$\sum_{p \in P_R} s_p \geq 2 \tag{2.5}$$

For all $p \in P_A, t \in p^\bullet$

$$e_{pt} \geq \frac{M(p)}{\psi(p)} \tag{2.6}$$

$$M(p) \geq e_{pt} \tag{2.7}$$

$$s_p + e_{pt} \leq 1 \tag{2.8}$$

$$\sum_{t \in T \setminus P^{0\bullet}} e_{pt} \leq |\{t \in T \setminus P^{0\bullet}\}| - 1 \tag{2.9}$$

For all $r \in P_R, t \in r^\bullet$

$$e_{rt} \geq \frac{M(r) - W(r,t) + 1}{M_0(r) - W(r,t) + 1} \tag{2.10}$$

$$\frac{M(r)}{W(r,t)} \geq e_{rt} \tag{2.11}$$

$$\sum e_{rt} + s_r \leq |r^\bullet| \tag{2.12}$$

For all $r, r' \in P_R, t \in r'^\bullet, r \in t^\bullet \cap P_R, p \in^\bullet t \cap P_A$

$$\begin{array}{l}(2s_{r'} - 1) \cdot M(r') \leq (2s_{r'} - 1) \cdot \{max[s_{r'} \cdot s_r \\ \cdot W(r',t)] - s_{r'}\}\end{array} \tag{2.13}$$

$$e_{r't} \cdot e_{pt} \cdot s_{r'} = 0 \tag{2.14}$$

$$s_p, e_{rt}, e_{pt} \in \{0, 1\} \tag{2.15}$$

$$M = M_0 + [N]Y, M \geq 0, Y \geq 0 \tag{2.16}$$

where $\psi(p)$ *can be directly obtained from the definition of a GS*^{3}*PR.*

The minimal non-max**-marked siphon is the set of places whose associated variables* s_p*'s are 1.*

Proof : Let us first make some comments on the variables used in the constraints.

Constraints (2)–(5) : Constraint (2) ensures that s is the characteristic vector of siphon S. Constraints (3)–(5) guarantee that the solution obtained contains no idle place, at least an activity place and two resource places as shown in Theorem 1.4.

Constraints (6)–(9) : For each $t \in p^\bullet$, $p \in P_A$, e_{pt} indicates whether arc (p,t) is enabled. It follows immediately from the following facts :

• Since $\psi(p) > 0$, $M(p)/\psi(p) > 0$ if $M(p) > 0$, which is equivalent to $e_{pt} = 1$.

• $e_{pt} = 0$ if $M(p) = 0$.

• For a non-live net, there exists at least one transition whose input arc (p, t) is disabled, where $p \in P_A$.

Constraints (10)–(15) : For each $t \in r^\bullet, r \in P_R$, e_{rt} indicates whether arc (r, t) is enabled. It follows immediately from the following facts :

• If t is enabled by arc (r, t) at M, i.e., $M(r) \geq W(r, t)$, then $\frac{M(r)}{W(r,t)} \geq 1$ and $1 \geq \frac{M(r)-W(r,t)+1}{M_0(r)-W(r,t)+1} > 0$. Hence, the value of e_{rt} must be 1.

• If t is disabled by arc (r, t) at M, i.e., $M(r) < W(r, t)$, then $\frac{M(r)}{W(r,t)} < 1$ and $0 \geq \frac{M(r)-W(r,t)+1}{M_0(r)-W(r,t)+1}$. Hence, the value of e_{rt} must be 0.

• For each resource place $r \in P_R$, if r is in the solution, there exists at least one output arc of r which is disabled.

• If there exists an SMS in a GS^{3}PR, there necessarily exist two resources r' and r. For $t \in r'^\bullet$, $r' \in P_R$, if $t^\bullet \cap P_R \neq \emptyset$, let $t^\bullet \cap P_R = \{r\}$. Constraint (13) means that when $s_{r'} = 0$, $M(r') \geq 0$; when $s_{r'} = 1$ and $s_r = 1$, $M(r') \leq max\{W(r', t)\} - 1$. Hence, Constraint (13) corresponds to the Definition of the non-max*-marked siphon (Definition 2.9 *(ii)*).

• Only when $e_{r't}$, e_{pt}, and $s_{r'}$ are all equal to one, the result of their multiplication is equal to one. However, based on Definition 2.9 *(iii)*, $e_{r't} \cdot e_{pt} \cdot s_{r'} = 1$ implies that the siphon including r' is max*-marked at M. We try to construct an IP to identify a minimal non-max*-marked at M. Thus we need Constraint (15), $e_{r't} \cdot e_{pt} \cdot s_{r'} = 0$. When one of $e_{r't}$ and e_{pt} is equal to zero, it is non-deterministic that r' belongs to a non-max*-marked siphon or not. However, $e_{r't} \cdot e_{pt} \cdot s_{r'} = 0$ is true.

If the IP has a feasible solution, the solution is a minimal non-max*-marked siphon with a corresponding marking $M \in R_S(N, M_0)$. □

Theorem 2.10

Let (N, M_0) be a well-marked GS^{3}PR. It is live if the IP in Theorem 2.9 has no feasible solution.

Proof : For a well-marked GS^{3}PR, the fact that the IP in Theorem 2.9 has no feasible solution means that all siphons of the net are max*-controlled. By Theorem 2.8, it is live. □

$R(N, M_0) \subseteq R_S(N, M_0)$ is true since the state equation does not check the feasibility of a transition sequence. The markings in $R_S(N, M_0) \backslash R(N, M_0)$ are called spurious markings. Although the reachability set derived from the state equation may contain spurious markings, its linear description facilitates the liveness analysis of a GS^{3}PR.

The existence of spurious markings sometimes prevents Theorem 2.9 from being a necessary condition. It means that for a live GS^{3}PR, the IP model in Theorem 2.9 may still obtain a solution that pertains to a spurious marking. Note that a GS^{3}PR is live iff there is no feasible solution in $R(N, M_0)$ to this IP test. However, $R(N, M_0)$ usually cannot be represented by a linear constraint.

2.6 Examples and Discussions

Example 1 : Take the net shown in Fig. 2.1(a) as an example. Liveness is checked by solving the IP in Theorem 2.9. Let $s = [x_1, x_2, ..., x_{14}]^T$ and $M = [y_1, y_2, ..., y_{14}]^T$, where $x_i = 0$ or $x_i = 1$, $y_i \geq 0$, $i = 1, 2, ..., 14$. Specifically, we have $M_0 = 9p_9 + 9p_{10} + 3p_{11} + 11p_{12} + 10p_{13} + 2p_{14}$. We use Lingo [78] to solve the following IP problem :

$min = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10}$;

$x_{11} + x_9 - x_1 \geq 0$;

$2 * x_1 + 2 * x_{10} - x_2 - x_9 \geq 0$;

$2 * x_2 - x_{10} - x_{11} \geq 0$;

$x_{12} + x_9 - x_3 \geq 0$;

$2 * x_3 + 2 * x_{10} - x_4 - x_9 \geq 0$;

$2 * x_4 - x_{10} - x_{12} \geq 0$;

$x_{13} + x_{10} - x_5 \geq 0$;

$2 * x_5 + 2 * x_9 - x_6 - x_{10} \geq 0$;

$2 * x_6 - x_9 - x_{13} \geq 0$;

$x_{14} + x_{10} - x_7 \geq 0$;

$2 * x_7 + 2 * x_9 - x_8 - x_{10} \geq 0$;

$2 * x_8 - x_9 - x_{14} \geq 0$;

$x_{11} + x_{12} + x_{13} + x_{14} = 0$;

$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 \geq 1$;

$x_9 + x_{10} \geq 2$;

$5 * y_1 + 4 * y_3 + y_6 + 6 * y_8 + y_9 = 9$;

$4 * y_2 + y_4 + 6 * y_5 + 5 * y_7 + y_{10} = 9$;

$y_1 + y_2 + y_{11} = 3$;

$y_3 + y_4 + y_{12} = 11$;

$y_5 + y_6 + y_{13} = 10$;

$y_7 + y_8 + y_{14} = 2$;

$e_{t_2} - y_1 \geq 0$;

$2 * e_{t_3} - y_2 \geq 0$;

$2 * e_{t_5} - y_3 \geq 0$;

$9 * e_{t_6} - y_4 \geq 0$;

$e_{t_8} - y_5 \geq 0$;

$9 * e_{t_9} - y_6 \geq 0$;

$e_{t_{11}} - y_7 \geq 0$;

$e_{t_{12}} - y_8 \geq 0$;

$y_1 - e_{t_2} \geq 0$;

$y_2 - e_{t_3} \geq 0$;

$y_3 - e_{t_5} \geq 0$;

$y_4 - e_{t_6} \geq 0$;

$y_5 - e_{t_8} \geq 0$;

$y_6 - e_{t_9} \geq 0$;

$y_7 - e_{t_{11}} \geq 0$;

$y_8 - e_{t_{12}} \geq 0$;

$$x_1 + e_{t_2} \leq 1\,;$$

$$x_2 + e_{t_3} \leq 1\,;$$

$$x_3 + e_{t_5} \leq 1\,;$$

$$x_4 + e_{t_6} \leq 1\,;$$

$$x_5 + e_{t_8} \leq 1\,;$$

$$x_6 + e_{t_9} \leq 1\,;$$

$$x_7 + e_{t_{11}} \leq 1\,;$$

$$x_8 + e_{t_{12}} \leq 1\,;$$

$$e_{t_2} + e_{t_3} + e_{t_5} + e_{t_6} + e_{t_8} + e_{t_9} + e_{t_{11}} + e_{t_{12}} \leq 7\,;$$

$$5 * e_{r_9 t_1} - y_9 \geq -4\,;$$

$$6 * e_{r_9 t_4} - y_9 \geq -3\,;$$

$$9 * e_{r_9 t_8} - y_9 \geq 0\,;$$

$$4 * e_{r_9 t_{11}} - y_9 \geq -5\,;$$

$$6 * e_{r_{10} t_2} - y_{10} \geq -3\,;$$

$$9 * e_{r_{10} t_5} - y_{10} \geq 0\,;$$

$$4 * e_{r_{10} t_7} - y_{10} \geq -5\,;$$

$$5 * e_{r_{10} t_{10}} - y_{10} \geq -4\,;$$

$$y_9 - 5 * e_{r_9 t_1} \geq 0\,;$$

$$y_9 - 4 * e_{r_9 t_4} \geq 0\,;$$

$$y_9 - e_{r_9 t_8} \geq 0\,;$$

$$y_9 - 6 * e_{r_9 t_{11}} \geq 0\,;$$

$$y_{10} - 4 * e_{r_{10} t_2} \geq 0\,;$$

$$y_{10} - e_{r_{10} t_5} \geq 0\,;$$

$$y_{10} - 6 * e_{r_{10} t_7} \geq 0\,;$$

$$y_{10} - 5 * e_{r_{10} t_{10}} \geq 0\,;$$

$e_{r_9t_1} + e_{r_9t_4} + e_{r_9t_8} + e_{r_9t_{11}} + x_9 \leq 4$;

$e_{r_{10}t_2} + e_{r_{10}t_5} + e_{r_{10}t_7} + e_{r_{10}t_{10}} + x_{10} \leq 4$;

$(2 * x_9 - 1) * y_9 \leq (2 * x_9 - 1) * (@smax(x_9 * x_{10} * 1, x_9 * x_{10} * 6, 0) - x_9)$;

$(2 * x_{10} - 1) * y_{10} \leq (2 * x_{10} - 1) * (@smax(x_{10} * x_9 * 4, x_{10} * x_9 * 1, 0) - x_{10})$;

$e_{r_9t_8} * e_{t_8} * x_9 * x_{10} = 0$;

$e_{r_9t_{11}} * e_{t_{11}} * x_9 * x_{10} = 0$;

$e_{r_{10}t_2} * e_{t_2} * x_9 * x_{10} = 0$;

$e_{r_{10}t_5} * e_{t_5} * x_9 * x_{10} = 0$;

Note that the above source code follows the syntax of Lingo [78], where function @smax(f_1, f_2, ..., f_n) indicates that the maximal value of $f_1, f_2, ..., and\ f_n$ is returned. No feasible solution can be found by solving this programming problem. It means that this net is live.

Actually, there is a unique SMS in this live net : $S = \{p_2, p_4, p_6, p_8, p_9, p_{10}\}$ that is always sufficiently marked. However, the MIPs in both [125] and [81] can find a feasible solution for this net. Using the MIP in [125], the SMS S is non-max-marked at marking $M = p_3 + p_5 + 5p_9 + 3p_{10} + 3p_{11} + 10p_{12} + 9p_{13} + 2p_{14}$. Using the MIP in [81], a minimal siphon S is non-max$''$-marked at $M = p_1 + p_3 + p_5 + 3p_{10} + 2p_{11} + 10p_{12} + 9p_{13} + 2p_{14}$. As shown in Table 2.2, S in Fig. 2.1(a) is non-max-marked at eight markings and non-max$''$-marked at one marking. Hence, a solution can be found. The control policy in [125] adds a control place for the obtained SMS, which is not necessary by the method proposed in this current chapter, as shown via this example.

Example 2 : The net shown in Fig. 2.5 has the same structure with a net in [81] and [9]. Fig. 2.6 is the reachability graph of the net in Fig. 2.5. Clearly, we can see that this net is deadlock-free and states $M_9 = p_1 + p_2 + p_4 + p_5 + p_7 + p_{10}$ and $M_{10} = p_1 + p_2 + p_4 + p_5 + p_7 + p_8 + p_9$ are livelocks in Fig. 2.6. Here we use Theorem 2.9 to detect the minimal non-max*-marked siphon.

In Fig. 2.5, let $s = [x_1, x_2, ..., x_{10}]^T$ and $M = [y_1, y_2, ..., y_{10}]^T$, where $x_i = 0$ or $x_i = 1$, $y_i \geq 0$, $i = 1, 2, ..., 10$. Specifically, we have $M_0 = 2p_1 + 2p_4 + 3p_7 + 3p_8 + p_9$. The minimal non-max*-marked siphon can be detected by the following IP problem :

$min = x_2 + x_3 + x_5 + x_6 + x_7 + x_8 + x_{10}$;

$x_1 + x_7 - x_2 \geq 0$;

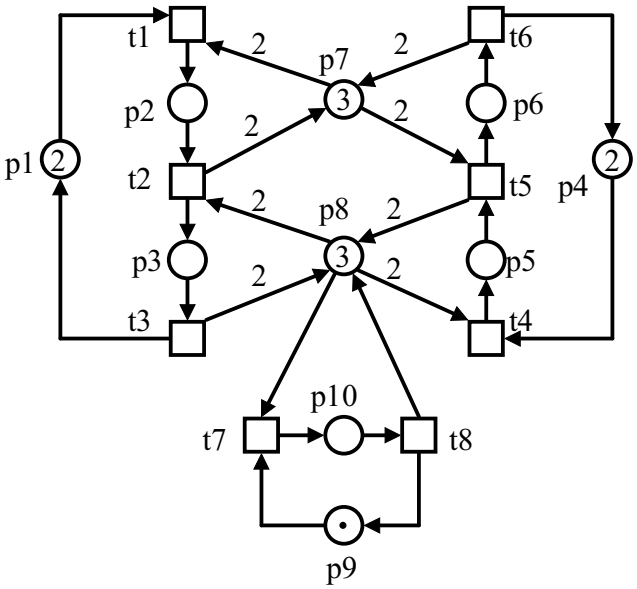


FIG. 2.5 – A net in [81].

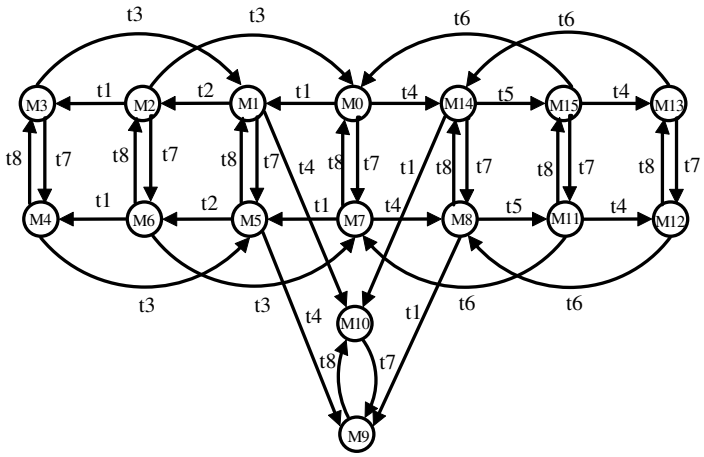


FIG. 2.6 – The reachability graph of the net in Fig. 2.5.

$2 * x_2 + 2 * x_8 - x_3 - x_7 \geq 0$;

$2 * x_3 - x_1 - x_8 \geq 0$;

$x_4 + x_8 - x_5 \geq 0$;

$2 * x_5 + 2 * x_7 - x_6 - x_8 \geq 0$;

$2 * x_6 - x_4 - x_7 \geq 0$;

$x_8 + x_9 - x_{10} \geq 0$;

$2 * x_{10} - x_8 - x_9 \geq 0$;

$x_1 + x_4 + x_9 = 0$;

$x_2 + x_3 + x_5 + x_6 + x_{10} \geq 1$;

$x_7 + x_8 \geq 1$;

$2 * y_2 + 2 * y_6 + y_7 = 3$;

$2 * y_3 + 2 * y_5 + y_8 + y_{10} = 3$;

$y_1 + y_2 + y_3 = 2$;

$y_4 + y_5 + y_6 = 2$;

$y_9 + y_{10} = 1$;

$e_{t_2} - y_2 \geq 0$;

$e_{t_3} - y_3 \geq 0$;

$e_{t_5} - y_5 \geq 0$;

$e_{t_6} - y_6 \geq 0$;

$e_{t_8} - y_{10} \geq 0$;

$y_2 - e_{t_2} \geq 0$;

$y_3 - e_{t_3} \geq 0$;

$y_5 - e_{t_5} \geq 0$;

$y_6 - e_{t_6} \geq 0$;

$y_{10} - e_{t_8} \geq 0$;

$x_2 + e_{t_2} \leq 1$;

$x_3 + e_{t_3} \leq 1$;

$x_5 + e_{t_5} \leq 1$;

$x_6 + e_{t_6} \leq 1$;

$x_{10} + e_{t_8} \leq 1$;

$e_{t_2} + e_{t_3} + e_{t_5} + e_{t_6} + e_{t_8} \leq 4$;

$2 * e_{r_7 t_5} - y_7 \geq -1$;

$2 * e_{r_8 t_2} - y_8 \geq -1$;

$2 * e_{r_8 t_4} - y_8 \geq -1$;

$3 * e_{r_8 t_7} - y_8 \geq 0$;

$2 * e_{r_7 t_1} - y_7 \geq -1$;

$y_7 - 2 * e_{r_7 t_1} \geq 0$;

$y_7 - 2 * e_{r_7 t_5} \geq 0$;

$y_8 - 2 * e_{r_8 t_2} \geq 0$;

$y_8 - 2 * e_{r_8 t_4} \geq 0$;

$y_8 - e_{r_8 t_7} \geq 0$;

$e_{r_7 t_1} + e_{r_7 t_5} + x_7 \leq 2$;

$e_{r_8 t_2} + e_{r_8 t_4} + e_{r_8 t_7} + x_8 \leq 3$;

$(2 * x_7 - 1) * y_7 \leq (2 * x_7 - 1) * (2 * x_7 * x_8 - x_7)$;

$(2 * x_8 - 1) * y_8 \leq (2 * x_8 - 1) * (2 * x_8 * x_7 - x_8)$;

$e_{r_7 t_5} * e_{t_5} * x_7 * x_8 = 0$;

$e_{r_8 t_2} * e_{t_5} * x_8 * x_7 = 0$;

Solving the above programming problem gives $min = x_2 + x_3 + x_5 + x_6 + x_7 + x_8 + x_{10} = 5$, where $x_3 = 1$, $x_6 = 1$, $x_7 = 1$, $x_8 = 1$, $x_{10} = 1$, and the others are zero. The corresponding minimal non-max*-marked siphon is $S = \{p_3, p_6, p_7, p_8, p_{10}\}$. Meanwhile, we can obtain the bad marking $M = p_1 + p_2 + p_4 + p_5 + p_7 + p_8 + p_9$. Contrasted with the reachability graph in Fig. 2.6, this

bad marking is M_{10}. For Fig. 2.5, t_7 fires once at M_{10}, $M_9 = p_1 + p_2 + p_4 + p_5 + p_7 + p_{10}$ is then obtained. Once the system evolves at M_9 or M_{10}, it can never process other processes. They are livelocks.

For this example, one cannot detect a deadly marked siphon by the MIP proposed in [124]. At bad markings M_9 and M_{10}, siphon $\{p_3, p_6, p_7, p_8, p_{10}\}$ is not deadly marked according to the definition of deadly marked siphons. Specifically, t_7 and t_8 are in the preset of the siphon. Transition t_8 is enabled by p_{10} at M_9 and t_7 is enabled by p_8 at M_{10}. These do not satisfy the condition $\forall t \in {}^\bullet S$, t is disabled by some $p \in S$.

In fact, compared with the MIP technique in [124], the proposed IP is more general in the two aspects : (i) a minimal non-max*-marked siphon can be obtained directly if a net is non-live and (ii) the new IP can solve the problem if a Petri net contains livelocks caused by siphons.

2.7 Summary

Deadlocks in an AMS can always be mapped into siphons of their Petri net models. They can be prevented by the proper control of siphons. Current deadlock control approaches suffer from restricted liveness characterization based on the controllability conditions of siphons. This chapter develops a necessary and sufficient controllability condition for GS^{3}PR. The contributions of this chapter consists of (1) proposal of a new controllability condition of siphons in GS^{3}PR, (2) development of a sufficient and necessary condition for the liveness of a GS^{3}PR, and (3) formulation of an IP model to detect the minimal non-max*-marked siphon that cause deadlocks or livelocks in GS^{3}PR. This chapter is another step towards a better knowledge about structural mechanisms ensuring a siphon to be controlled. This permits us to look forward a broader decision power of the controlled siphon property in particular for systems where the purely algebraic methods reach their limit. The use of the max*-controllability condition and the proposed IP model to control a generalized Petri net is still a problem requiring a further study. A sufficient and necessary siphon control condition for S^{4}R remains open. Also, it is challenging to design an optimal supervisor for generalized Petri nets based on siphons.

The major contributions in this research are published :

[1] **Gaiyun Liu** and Kamel Barkaoui, Necessary and sufficient liveness condition of GS^{3}PR Petri

nets, *International Journal of Systems Science*, DOI : 10.1080/00207721.2013.827257(online), 2013.

[2] **Gaiyun Liu**, Zhiwu Li, Abdulrahman M. Al-Ahmari, Liveness analysis of Petri nets using siphons and mathematical programming, *12th IFAC International Workshop on Discrete Event Systems in Paris*, 2014.

Chapitre 3

Deadlock Prevention for M-nets Based on Structure Reuse of Supervisors

3.1 Introduction

The existing prevention policies underlying Petri net formalisms are developed on the basis of either a state space or structural analysis, e.g., siphon control. Falling into the first category, the theory of regions that can derive Petri nets from automaton-based models is an important method for supervisory control of discrete event systems. The most attractive advantage of the approach is that an optimal supervisor can always be obtained, by adding monitors that are used to separate events from unsafe states, when such a supervisor exists. However, it bears much computational cost. One first needs to generate the reachability graph given a Petri net model. Then, the set of marking/transition separation instances is found, whose number is in theory exponential with respect to the net size and initial marking. Finally, for each instance, a monitor is found by solving a linear programming problem in which the number of constraints is approximately equal to that of nodes in the reachability graph. When its initial marking changes, the aforementioned steps have to be repeated. In such an approach, no information of previously determined supervisors can be reused.

Deadlock prevention based on siphon control is a typical application of structural analysis techniques of Petri nets. It is not optimal and can even be overly restrictive and conservative in many cases. However, it is computationally tractable and allows its supervisor to be reused when a system experiences such changes as new capacity and job instances [23], [24], [46], [47], [48],

[69], [70], [100], [122], [123].

To inherit and preserve the advantages of two classes of the approaches, this chapter proposes a novel design method of deadlock prevention supervisors based on Petri nets, which does not guarantee optimality but empirical results show its superiority over other approaches based on siphon control. Given the Petri net model of an AMS, one first designs an optimal liveness-enforcing controlled system for the model at a minimal initial marking by utilizing the theory of regions. Then, we calculate all SMS in the controlled system. Such a siphon does not contain a trap. For each SMS, an algebraic inequality with respect to the markings of monitors and resource places in the controlled system, also called a liveness constraint, is established in terms of the concept of max-controlled or invariant-controlled siphons. Its satisfaction implies the absence of dead transitions in the postset of the corresponding siphon. Consequently, given initial markings that satisfy all the liveness inequality constraints, all siphons can be max-controlled, and the resulting controlled system is live.

After a controlled system structure is found, one can reallocate the initial markings according to the inequality constraints. No matter how large the initial markings and the number of states are, the liveness constraints remain unchanged. Their satisfaction ensures the absence of uncontrolled siphons. This implies that, for a plant model with a fixed net structure, we only need to compute its reachability graph at a minimal initial marking and the siphons of the controlled system once. Whenever the number of process instances and the capacity of manufacturing resources change, a Petri net supervisor can be determined easily via these algebraic inequality constraints.

The remainder of this chapter is organized as follows. Section 2 formulates the considered problem through a motivation example and generalizes a class of manufacturing-oriented Petri nets. Section 3 elaborates a method that properly allocates initial markings for monitors to prevent siphons from being uncontrolled. Section 4 proposes an algorithm to identify the redundant constraints. A deadlock prevention policy is developed in Section 5. AMS examples are given in Section 6, showing the near optimality achieved by the proposed method. A problem of the proposed method is discussed in Section 7. Finally, Section 8 concludes the chapter and identifies research directions for future work.

3.2 Structure Design of a Petri Net Supervisor

3.2.1 Motivation and Problem Formulation

Let us recall the steps to use the theory of regions to design a supervisor for a Petri net model. First, we generate the reachability graph of the model. Then, all marking/transition separation instances are found. For each instance, a monitor is computed by solving a linear programming problem (LPP). In theory, the number of marking/transition separation instances grows exponentially with the net size and initial marking. So is the number of constraints in each LPP. Moreover, the size of a reachability graph is rather sensitive to the size and initial marking of a net. These facts make it infeasible for the theory of regions to be applied to real-world problems.

We formulate the problem and illustrate the proposed method through a small example from [130]. Consider a net (N, M_0) in Fig. 3.1(a) with its reachability graph shown in Fig. 3.1(b). Its optimal controlled system (N^c, M_0^c) can be found by the theory of regions [104], [38], as shown in Fig. 3.1(f). Now we consider the deadlock control in (N^m, M_0^m) as shown in Fig. 3.1(c). It has the same topology structure as (N, M_0) in Fig. 3.1(a), but has a small initial marking. Its reachability graph is shown in Fig. 3.1(d). Fig. 3.1(e) shows its controlled system (N^{mc}, M_0^{mc}) obtained by using the theory of regions. Finding (N^{mc}, M_0^{mc}) is obviously more tractable than finding (N^c, M_0^c) since (N^m, M_0^m) has a small reachability space.

Now we investigate the relationship between the controllability of siphons in (N^{mc}, M_0^{mc}) and its initial marking. N^{mc} has five minimal siphons : $S_1 = \{p_1, p_2, p_3, p_4\}$, $S_2 = \{p_3, p_5\}$, $S_3 = \{p_2, p_4, p_6\}$, $S_4 = \{p_2, p_3, p_c\}$, and $S_5 = \{p_4, p_5, p_6\}$. The first four are also traps, implying that they cannot be unmarked once p_1, p_5, p_6, and p_c are initially marked. From the original model (N, M_0), p_1, p_5, and p_6 are initially marked. As a monitor, p_c must be initially marked. Otherwise there exist dead transitions at the initial marking. Next we give a marking relation with which S_5 is controlled.

Note that $I_{p_5} = p_3 + p_5$, $I_{p_6} = p_2 + p_4 + p_6$, and $I_{p_c} = p_2 + p_3 + p_c$ are P-invariants in Fig. 3.1(e). Let $I = I_{p_5} + I_{p_6} - I_{p_c}$. Clearly, $I = p_4 + p_5 + p_6 - p_c$ is a P-invariant. Since $\|I\|^+ \subseteq S_5$, S_5 is controlled if $I^T M_0^{mc} > 0$, i.e., $M_0^{mc}(p_4) + M_0^{mc}(p_5) + M_0^{mc}(p_6) > M_0^{mc}(p_c)$. The above results indicate that each siphon in (N^m, M_0^m) is controlled if p_1, p_5, and p_6 are initially marked, and $M_0^{mc}(p_4) + M_0^{mc}(p_5) + M_0^{mc}(p_6) > M_0^{mc}(p_c)$.

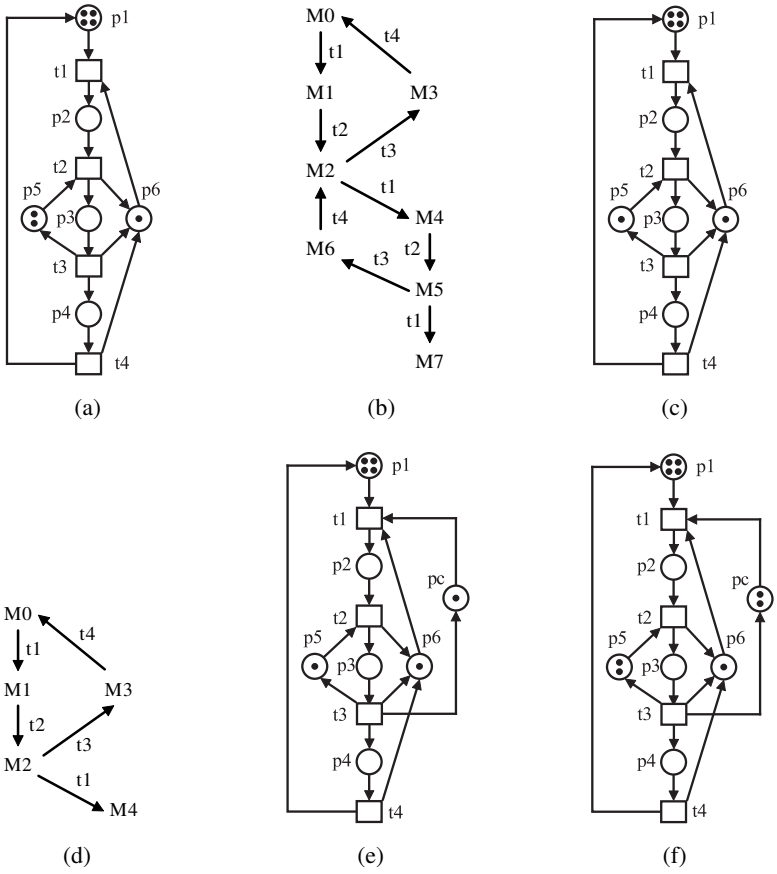


FIG. 3.1 – (a) A plant model (N, M_0), (b) the reachability graph of (N, M_0), (c) a modified model (N^m, M_0^m), (d) the reachability graph of (N^m, M_0^m), (e) a controlled system (N^{mc}, M_0^{mc}) for (N^m, M_0^m), (f) a controlled system (N^c, M_0^c) for (N, M_0).

Now we consider the deadlock prevention problem for (N, M_0) by using the structure of the controlled system in Fig. 3.1(e). In (N, M_0), $M_0 = 4p_1 + 2p_5 + p_6$, i.e., p_1, p_5, and p_6 are initially marked. If p_c is initially marked and $M_0^c(p_5) + M_0^c(p_6) > M_0^c(p_c)$, then a controlled system for (N, M_0) can be obtained. Since $M_0^c(p_5) = M_0(p_5) = 2$ and $M_0^c(p_6) = M_0(p_6) = 1$, $M_0^c(p_c) = 2$ means the truth of $M_0^c(p_5) + M_0^c(p_6) > M_0^c(p_c)$, as shown in Fig. 3.1(f).

In summary, by using the structure of the controlled system of a net with a small initial marking, we can compute a controlled system for the same net structure with a large initial marking, in terms of algebraic inequality constraints with respect to markings. That is to say, once the structure of a controlled system is determined, the initial marking of monitors is determined by a set of inequality constraints. For a plant model (N, M_0') with its structure shown in Fig. 3.1(a) and a new initial marking $M_0' = 4p_1 + 3p_5 + p_6$, one can easily find a controlled system with its structure shown in Fig. 3.1(f) and $M_0^{mc}(p_c) = 3$ such that $M_0^c(p_5) + M_0^c(p_6) > M_0^c(p_c)$ is true. It is not necessary to apply the theory of regions afresh to (N, M_0').

Let (N, M_0) be a plant net model with place set P and P_V be the set of monitors in its controlled system (N^c, M_0^c). The deadlock prevention procedure proposed in this study contains the following steps :

(1) Find a controlled system (N^{mc}, M_0^{mc}) for (N^m, M_0^m) by using the theory of regions, where $N^m = N$ and $M_0^m \leq M_0$.

(2) Derive the controllability conditions of siphons in N^{mc}, which are represented by algebraic inequalities of markings of the places in the plant model and monitors in N^{mc}.

(3) Decide an initial marking M_0^c such that $\forall p \in P$, $M_0^c(p) = M_0(p)$ and $\forall p \in P_V$, $M_0^c(p)$ satisfies its corresponding inequality constraints. (N^c, M_0^c) is a controlled system for (N, M_0), where $N^c = N^{mc}$.

3.2.2 M-nets

This study considers deadlock problems for a class of manufacturing-oriented Petri nets, M-nets for short. It is a generalization of the existing net classes that model AMS.

Definition 3.1 *An M-net denoted by* (N, M_0) *satisfies the following statements :*

1. $N = \bigcirc_{i=1}^{n} N_i = (P^0 \cup P_A \cup P_R, T, F, W)$ *is composed of n nets* N_1, N_2, ..., *and* N_n, *where* $\forall i \in \mathbb{N}_n$, $N_i = (\{p_i^0\} \cup P_{A_i} \cup P_{R_i}, T_i, F_i, W_i)$ *is called a subnet of* N.
2. $P^0 = \cup_{i=1}^{n} \{p_i^0\}$ *is called a set of idle process places with* $p_i^0 \neq p_j^0$, $\forall i, j \in \mathbb{N}_n$, $i \neq j$; $P_A = \cup_{i=1}^{n} P_{A_i}$ *is called a set of activity places with* $P_{A_i} \cap P_{A_j} = \emptyset$, $\forall i, j \in \mathbb{N}_n$, $i \neq j$; *and* $P_R = \cup_{i=1}^{n} P_{R_i}$ *is called a set of resource places.*
3. $\forall i, j \in \mathbb{N}_n$, $i \neq j$, $T_i \cap T_j = \emptyset$.
4. $\forall r \in P_R$, *it is associated with a minimal P-semiflow* I_r *such that* $I_r(r) = 1$, $\forall p \in P_A$, $I_r(p) \geq 0$, *and* $\forall p \in P^0$, $I_r(p) = 0$.
5. $\forall p \in P_A$, p *is associated with a minimal P-semiflow* I_p, *where* $\|I_p\| \subseteq P_A$.
6. (N_i, M_{0i}) *is quasi-live, bounded, and conservative.*
7. (N_i', M_{0i}') *with* $N_i' = (\{p_i^0\} \cup P_{A_i}, T_i, F_i', W_i')$ *is live, bounded, and reversible, where* N_i' *is the resulting net from removing resource places in* (N_i, M_{0i}).
8. *Let* (N_i, M_{0i}) $(i = 1, 2)$ *be two subnets with* $N_i = (\{p_i^0\} \cup P_{A_i} \cup P_{R_i}, T_i, F_i, W_i)$. *Their composition, denoted by* (N_{12}, M_{12}) *with* $N_{12} = N_1 \circ N_2 = (P_{12}^0 \cup P_{A_{12}} \cup P_{R_{12}}, T_{12}, F_{12}, W_{12})$, *is defined as follows :*
   - $P_{12}^0 = \{p_1^0\} \cup \{p_2^0\} = \{p_1^0, p_2^0\}$, $P_{A_{12}} = P_{A_1} \cup P_{A_2}$, *and* $P_{R_{12}} = P_{R_1} \cup P_{R_2}$
   - $T_{12} = T_1 \cup T_2$
   - $F_{12} = F_1 \cup F_2$
   - $\forall f \in F_1$, $W(f) = W_1(f)$ *and* $\forall f \in F_2$, $W(f) = W_2(f)$
   - $\forall p \in \{p_1^0\} \cup P_{A_1}$, $M_{12}(p) = M_{01}(p)$; $\forall p \in \{p_2^0\} \cup P_{A_2}$, $M_{12}(p) = M_{02}(p)$; $\forall r \in P_{R_1} \setminus P_{R_2}$, $M_{12}(r) = M_{01}(r)$; $\forall r \in P_{R_2} \setminus P_{R_1}$, $M_{12}(r) = M_{02}(r)$; *and* $\forall r \in P_{R_1} \cap P_{R_2}$, $M_{12}(r) = max\{M_{01}(r), M_{02}(r)\}$
9. *The net* N *resulting from the composition of n subnets* N_1, N_2, ..., *and* N_n *is defined as follows : if* $n = 1$, *then* $N = N_1$; *if* $n > 1$, *then* $N = \bigcirc_{i=1}^{n} N_i = (\bigcirc_{i=1}^{n-1} N_i) \circ N_n$.
10. $\forall p \in P^0$, $M_0(p) > 0$; $\forall p \in P_A$, $M_0(p) = 0$; *and* $\forall r \in P_R$, $M_0(r) \geq max\{I_r(p) | p \in \|I_r\|\}$. *Such a marking is said to be an admissible initial marking.*
11. *An uncontrolled siphon in* (N, M_0) *contains at least one resource place and one activity place but no idle process place.*
12. (N, M_0) *is live if no siphon is uncontrolled.*

13. *Liveness can be enforced to* (N, M_0) *by adding monitors whose addition leads to a controlled system.*

14. *Let* (N^c, M_0^c) *be a controlled system for* (N, M_0). (N^c, M_0^c) *is live if it is ordinary and no siphon is unmarked.* (N^c, M_0^c) *is live if it is generalized and satisfies the cs-property.*

15. *Let* P_V *be the set of monitors in* (N^c, M_0^c). $\forall v \in P_V$, *there exists a minimal P-semiflow* I_v *such that* $I_v(v) = 1$ *and* $\forall p \in \|I_v\| \setminus \{v\}$, $p \in P_A$.

For example, the net shown in Fig. 3.1(a) is an M-net, where p_1 is an idle process place, p_2, p_3, and p_4 are activity places, and p_5 and p_6 are resource places. It is quasi-live, bounded, and conservative. It is live if no siphon is uncontrolled. For example, the net at initial marking $2p_1 + 2p_5 + p_6$ is live since every siphon can never be emptied.

It is easy to verify that M-nets are more general than almost all manufacturing-oriented Petri net subclasses in the literature such as PPN, augmented marked graphs [16], S^3PR [23], L-S^3PR [24], S^4R [81], S^4PR [100], ES^3PR [47], WS^3PSR [99], S^*PR [25], S^3PMR [48], PNR [54], RCN-merged nets [53], ERCN-merged nets [121], ERCN*-merged nets [55], S^3PGR^2 [89], G-tasks [5], and well-formed G-systems [133]. In [77], a formal proof is presented to show that an M-net is more general than a well-formed G-system.

3.2.3 Minimal Initial Marking

Let (N, M_0) be a plant M-net in which M_0 is admissible. This section finds a minimal initial marking M_0^m at which M-net N contains deadlocks and any SMS of N can become uncontrolled at a marking $M \in R(N, M_0^m)$.

Algorithm 3.1

finding a minimal initial marking for N

Input : a plant model (N, M_0) *with* $N = (P^0 \cup P_A \cup P_R, T, F, W)$

Output : M_0^m, *"*(N, M_0) *is live", or "*(N, M_0) *cannot be handled by the proposed method"*

begin{

the MIP-based deadlock detection method [16], [89] is applied to (N, M_0)

if *{there are uncontrolled siphons in* (N, M_0)*}* ***then***

compute the set Π_u *of strict minimal siphons in* N

$\forall p \in P^0 \cup P_A$, $M_0^m(p) := M_0(p)$

$\forall r \in P_R$, $M_0^m(r) := max\{I_r(p) | p \in \|I_r\|\}$

the MIP-based deadlock detection method is applied to (N, M_0^m)

if *{there are no uncontrolled siphons in* (N, M_0^m)*}* ***then***

flag :=2

else

compute the set of reachable markings of (N, M_0^m)

Find the set of dead markings $R_D(N, M_0^m)$

if *{*$\forall S \in \Pi_u$, $\exists M \in R_D(N, M_0^m)$, S *is uncontrolled at* M*}* ***then***

flag :=0

else

flag :=2

end if

end if

else

flag :=1

end if

if *{flag==2}* ***then***

output "(N, M_0) *cannot be handled by the proposed method"*

else

if *{flag==1}* ***then***

output "(N, M_0) *is live"*

else

output M_0^m

end if

end if

}end of the algorithm

First, Algorithm 3.1 decides whether (N, M_0) is live. By the definition of M-nets, it is live if there is no uncontrolled siphon, which can be determined by an MIP-based deadlock detection

method. If it is live (flag=1), the algorithm exits. If it is not live, then the set of SMS and the minimal admissible initial marking M_0^m are computed. If (N, M_0^m) is live (flag=2), then we cannot find a minimal initial marking for it and its deadlock problems can be handled by any existing approach, e.g., the theory of regions and siphon-based ones. If (N, M_0^m) is not live and for any SMS, there exists a dead marking in $R(N, M_0^m)$ at which the SMS is uncontrolled, then the minimal admissible marking M_0^m is obtained. Note that this algorithm needs to compute the state space and the set of SMS of (N, M_0^m) if the original model is not live. However, if the algorithm outputs the minimal initial marking M_0^m for N, the information of the state space and siphons of (N, M_0^m) will be used later. Take the net (N, M_0) shown in Fig. 3.1(a) as an example. (N, M_0) is not live and $S = \{p_4, p_5, p_6\}$ is unique SMS. Let $M_0^m = 4p_1 + p_5 + p_6$. (N, M_0^m) is not live and S is uncontrolled at $M_4 = 2p_1 + p_2 + p_3$. Then $M_0^m = 4p_1 + p_5 + p_6$ is the minimal initial marking of (N, M_0).

3.2.4 Derivation of the Structure of a Controlled System

Let (N, M_0) be an M-net with $N = (P^0 \cup P_A \cup P_R, T, F, W)$. To find its controlled system, as stated in Section 3.2.1, we first design a controlled system for (N^m, M_0^m), where $N^m = N$ and M_0^m is the minimal initial marking.

Algorithm 3.2

structure design of a controlled system for (N, M_0)

Input : a plant model (N, M_0)

Output : (N^{mc}, M_0^{mc})

begin{

$N^m := N$

find the minimal initial marking M_0^m *by Algorithm 3.1*

if *{there exists an optimal controlled system for* (N^m, M_0^m)*}* ***then***

design a controlled system (N^{mc}, M_0^{mc}) *for* (N^m, M_0^m) *by the theory of regions*

else

design a controlled system (N^{mc}, M_0^{mc}) *for* (N^m, M_0^m) *by the method in [106]*

end if

output (N^{mc}, M_0^{mc})

}end of the algorithm

The motivation to design a controlled system (N^{mc}, M_0^{mc}) for (N^m, M_0^m) is that M_0^m is not greater than M_0 and it is more tractable by using the theory of regions to design a controlled system for (N^m, M_0^m) than that for (N, M_0).

The net shown in Fig. 3.1(a) with an initial marking $M_0 = 150p_1 + 100p_5 + 50p_6$ has more than 1.3×10^5 states. One can imagine the computational overhead if the theory of regions is applied to such a net. However, Algorithm 3.2 considers (N^m, M_0^m) as shown in Fig. 3.1(c), which has five reachable markings only. As a result, it is easy to compute a controlled system for (N^m, M_0^m) by using the theory of regions, as shown in Fig. 3.1(e).

3.3 Siphon Controllability Constraints

This section presents a set of algebraic inequality constraints with respect to the markings of resource places and monitors at which (N^{mc}, M_0^{mc}) is live as the output of Algorithm 3.1, where (N^{mc}, M_0^{mc}) is a controlled system for (N^m, M_0^m) with $N^{mc} = (P^0 \cup P_A \cup P_R \cup P_V, T, F^{mc}, W^{mc})$ and $N^m = (P^0 \cup P_A \cup P_R, T, F, W)$. Let Π_u denote the set of uncontrolled minimal siphons in (N^{mc}, M_0^{mc}). Π_u can be further divided into three disjoint subsets Π_G, Π_H, and Π_V, called the sets of plant, hybrid, and monitor siphons, respectively, such that $\forall S \in \Pi_G$, $S \subset P_A \cup P_R$; $\forall S \in \Pi_V$, $S \subset P_A \cap P_V$; and $\forall S \in \Pi_H$, $\exists r \in P_R$ and $v \in P_V$ such that $\{r, v\} \subseteq S$.

The net shown in Fig. 3.2(b) is an optimal controlled system for a plant model (N, M_0) depicted in Fig. 3.2(a) that is an M-net with $P^0 = \{p_1, p_5\}$, $P_R = \{p_9, p_{10}, p_{11}\}$, and the others are activity places. There are 15 minimal siphons : $S_1 = \{p_3, p_8, p_9, p_{10}\}$, $S_2 = \{p_4, p_7, p_{10}, p_{11}\}$, $S_3 = \{p_4, p_8, p_9, p_{10}, p_{11}\}$, $S_4 = \{p_3, p_7, v_1, v_2\}$, $S_5 = \{p_3, p_8, p_9, v_1, v_2\}$, $S_6 = \{p_4, p_7, p_{11}, v_1, v_2\}$, $S_7 = \{p_4, p_8, p_9, p_{11}, v_1, v_2\}$, $S_8 = \{p_1, p_2, p_3, p_4\}$, $S_9 = \{p_5, p_6, p_7, p_8\}$, $S_{10} = \{p_2, p_8, p_9\}$, $S_{11} = \{p_3, p_7, p_{10}\}$, $S_{12} = \{p_4, p_6, p_{11}\}$, $S_{13} = \{p_2, p_7, v_1\}$, $S_{14} = \{p_3, p_6, v_2\}$, and $S_{15} = \{p_2, p_6, v_3\}$. Note that $S_8 - S_{15}$ are also traps that are marked at the minimal initial marking by the definition of M-nets. As a result, they do not contribute to deadlocks. We have $\Pi_u = \{S_1, S_2, \ldots, S_7\}$ with $\Pi_G = \{S_1, S_2, S_3\}$, $\Pi_V = \{S_4\}$, and $\Pi_H = \{S_5, S_6, S_7\}$. $S \in \Pi_G$ can be represented by $S_A \cup S_R$, where $S_A \subseteq P_A$ and $S_R \subseteq P_R$.

Definition 3.2 *Given $S \in \Pi_G$ in N^{mc}, I_r is the minimal P-semiflow associated with $r \in P_R \cup P_V$, $\Omega_S = \sum_{r \in S_R} I_r$, and $\Omega'_S = \sum_{p \in S} \Omega_S(p)p$. Multiset $\bar{S} = \Omega_S - \Omega'_S$ is called S's complementary set.*

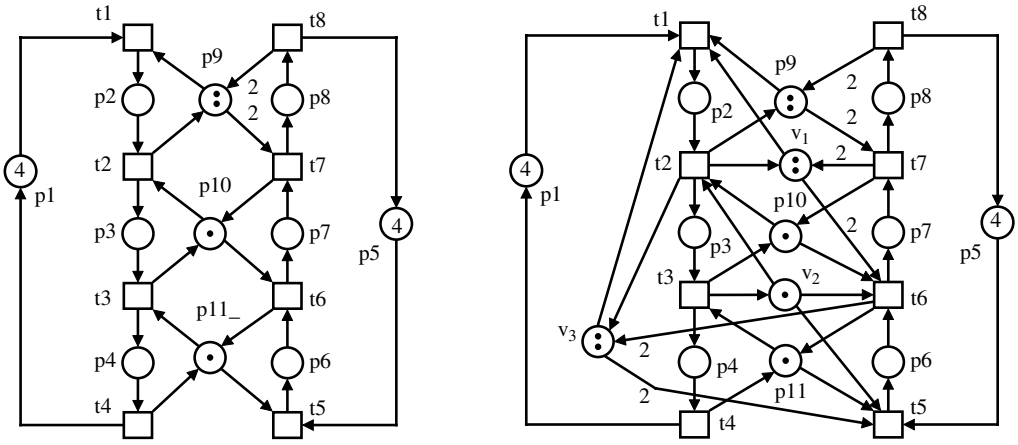


FIG. 3.2 – (a) A plant model (N, M_0), (b) a controlled system (N^{mc}, M_0^{mc}) for (N, M_0).

For example, $S_1 = \{p_3, p_8, p_9, p_{10}\} \in \Pi_G$ is a plant siphon in Fig. 3.2(b). We have $\Omega_{S_1} = p_2+2p_8+p_9+p_3+p_7+p_{10}$ and $\Omega'_{S_1} = p_3+2p_8+p_9+p_{10}$. Thus, $\bar{S_1} = p_2+p_7$. $S_2 = \{p_4, p_7, p_{10}, p_{11}\}$ is also a plant siphon with $\Omega_{S_2} = p_3 + p_7 + p_{10} + p_4 + p_6 + p_{11}$ and $\Omega'_{S_2} = p_4 + p_7 + p_{10} + p_{11}$. Therefore, we have $\bar{S_2} = p_3 + p_6$.

Definition 3.3 *Given $r \in P_R \cup P_V$, $H(r) = I_r - r$ is called the set of holders of r, where $H(r)$ is a multiset.*

Corollary 3.1

$\forall r \in P_R \cup P_V, H(r) \subseteq P_A$.

For example, $H(p_9) = I_{p_9} - p_9 = p_2 + 2p_8 + p_9 - p_9 = p_2 + 2p_8$ and $H_{p_{10}} = p_3 + p_7$. It is easy to verify that $\{p_2, p_8\} \subset P_A$ and $\{p_3, p_7\} \subset P_A$.

Definition 3.4 *Given $S \in \Pi_G$, $\alpha_S \subseteq P_R \cup P_V$ is called a minimal set of constraint places of S with respect to $A_S = \{d_{v_i} \in \mathbb{N}^+ | i = 1, 2, \ldots, |\alpha_S|\}$ if (1) $\alpha_S \neq S \setminus P_A$; (2) $\bar{S} \leq \sum_{v_i \in \alpha_S} d_{v_i} H(v_i)$; (3) $\exists p \in \bar{S}$, $\bar{S}(p) > \Omega(p)$, where $\Omega = \sum_{v_i \in \alpha_S} (d_{v_i} - 1)H(v_i)$; and (4) $\nexists \alpha'_S \subseteq P_R \cup P_V$ such that (1)–(3) are satisfied and $|\alpha'_S| < |\alpha_S|$.*

The addition of the sets of the holders of the monitors in α_S is greater than or equals to multiset $\bar{S}$. For example, $S_1 = \{p_3, p_8, p_9, p_{10}\}$ is a plant siphon in Fig. 3.2(b) with $\bar{S_1} = p_2 + p_7$. Note that $H(v_1) = p_2 + 2p_7$. We have $\alpha_{S_1} = \{v_1\}$ and $d_{v_1} = 1$ since $\bar{S_1} < p_2 + 2p_7$. Consider a plant siphon

$S_3 = \{p_4, p_8, p_9, p_{10}, p_{11}\}$ with $\bar{S_3} = p_2 + p_3 + p_6 + p_7$. We have $\alpha_{S_3} = \{v_1, v_2\}$ with $d_{v_1} = 1$ and $d_{v_2} = 1$.

Proposition 3.1

[4] Let (N, M_0) be a Petri net and S be a siphon of N. If there exists a P-invariant I such that $\forall p \in (\|I\|^- \cap S)$, $max_{p^\bullet} = 1$, $\|I\|^+ \subseteq S$, and $I^T M_0 > \sum_{p \in S} I(p)(max_{p^\bullet} - 1)$, then S is max-controlled.

Proposition 3.2

Given $S \in \Pi_G$ and $I_S = \sum_{r \in S_R} I_r - \sum_{v \in \alpha_S} d_v I_v$, S is max-controlled if $M_0^{mc}(S_R) - \sum_{v \in \alpha_S} d_v M_0^{mc}(v) > \sum_{p \in S} I_S(p)(max_{p^\bullet} - 1)$.

Proof : Both $\sum_{r \in S_R} I_r$ and $\sum_{v \in \alpha_S} d_v I_v$ are P-semiflows. As a result, I_S is a P-invariant. By the definition of α_S, we have $\bar{S} \leq \sum_{v \in \alpha_S} d_v H(v)$. Therefore, $\|I_S\|^- \cap S = \emptyset$, $\|I_S\|^+ \subseteq S$, and $\forall p \in \|I_S\| \setminus (S_R \cup \alpha_S)$, $M_0^{mc}(p) = 0$. $M_0^{mc}(S_R) - \sum_{v \in \alpha_S} d_v M_0^{mc}(v) > \sum_{p \in S} I_S(p)(max_{p^\bullet} - 1)$ implies the truth of $I_S^T M_0^{mc} > \sum_{p \in S} I_S(p)(max_{p^\bullet} - 1)$. By Proposition 3.1, S is max-controlled.

□

Note that the condition in Proposition 3.2 is rather conservative. This is due to Proposition 3.1. Consider $S_3 = \{p_4, p_8, p_9, p_{10}, p_{11}\}$ in Fig. 3.2(b). We have $\alpha_{S_3} = \{v_1, v_2\}$ with $d_{v_1} = 1$ and $d_{v_2} = 1$. As a result, S_3 is max-controlled if $M_0^{mc}(p_9) + M_0^{mc}(p_{10}) + M_0^{mc}(p_{11}) - M_0^{mc}(v_1) - M_0^{mc}(v_2) > 1$. It is easy to verify that the above inequality does not hold in Fig. 3.2(b). However, transitions in $S_3^\bullet$ are still live.

Corollary 3.2

Given S being a minimal siphon in an ordinary net N, $\forall p \in S$, $max_{p^\bullet} - 1 = 0$.

Proof : By contradiction, suppose that $\exists p \in S$, $p^\bullet = \emptyset$, it means that ${}^\bullet(S \setminus \{p\}) \subseteq (S \setminus \{p\})^\bullet$. This contradicts the minimality of siphon S. □

Corollary 3.3

Given $S \in \Pi_G$ in (N^{mc}, M_0^{mc}) that is ordinary, S is controlled if $M_0^{mc}(S_R) - \sum_{v \in \alpha_S} d_v M_0^{mc}(v) > 0$.

Proof : It immediately follows from Proposition 3.2 and Corollary 3.2. □

For example, $S_5 = \{p_4, p_5, p_6\}$ is a plant siphon in Fig. 3.1(e) that is ordinary. We have $\bar{S_5} = p_2 + p_3$ and $H(p_c) = p_2 + p_3$. Thus, $\alpha_{S_5} = \{p_c\}$. Note that $p_4 \in P_A$ and $\{p_5, p_6\} \subseteq P_R$. S_5 is controlled if $M_0^{mc}(p_5) + M_0^{mc}(p_6) - M_0^{mc}(p_c) > 0$.

Definition 3.5 *Given $S \in \Pi_V$, $\alpha_S \subseteq P_R \cup P_V$ is called a minimal set of constraint places of S with respect to $A_S = \{d_{v_i} \in \mathbb{N}^+ | i = 1, 2, \ldots, |\alpha_S|\}$ if (1) $\alpha_S \neq S \setminus P_A$; (2) $\bar{S} \leq \sum_{v_i \in \alpha_S} d_{v_i} H(v_i)$; (3) $\exists p \in \bar{S}$, $\bar{S}(p) > \Omega(p)$, where $\Omega = \sum_{v_i \in \alpha_S} (d_{v_i} - 1) H(v_i)$; and (4) $\nexists \alpha'_S \subseteq P_R \cup P_V$ such that (1)-(3) are satisfied and $|\alpha'_S| < |\alpha_S|$.*

Proposition 3.3

Given $S \in \Pi_V$ and $I_S = \sum_{v \in S \cap P_V} I_v - \sum_{r \in \alpha_S} d_r I_r$, S is max-controlled if
$M_0^{mc}(S \cap P_V) - \sum_{r \in \alpha_S} d_r M_0^{mc}(r) > \sum_{p \in S} I_S(p)(max_{p^\bullet} - 1)$.

For instance, $S_4 = \{p_3, p_7, v_1, v_2\}$ is a monitor siphon in Fig. 3.2(b). Its minimal set of constraint places is $\alpha_{S_4} = \{p_9, p_{11}\}$ with $H(p_9) = p_2 + 2p_8$, $d_{p_9} = 1$, $H(p_{11}) = p_4 + p_6$, and $d_{p_{11}} = 1$. Thus, S_4 is max-controlled if $M_0^{mc}(v_1) + M_0^{mc}(v_2) - M_0^{mc}(p_9) - M_0^{mc}(p_{11}) > 1$.

Definition 3.6 *Given $S \in \Pi_H$, $\alpha_S \subseteq P_R \cup P_V$ is called a minimal set of constraint places of S with respect to $A_S = \{d_{v_i} \in \mathbb{N}^+ | i = 1, 2, \ldots, |\alpha_S|\}$ if (1) $\alpha_S \neq S \setminus P_A$; (2) $\bar{S} \leq \sum_{v_i \in \alpha_S} d_{v_i} H(v_i)$; (3) $\exists p \in \bar{S}$, $\bar{S}(p) > \Omega(p)$, where $\Omega = \sum_{v_i \in \alpha_S} (d_{v_i} - 1) H(v_i)$; and (4) $\nexists \alpha'_S \subseteq P_R \cup P_V$ such that (1)–(3) are satisfied and $|\alpha'_S| < |\alpha_S|$.*

Proposition 3.4

Given $S \in \Pi_H$ and $I_S = \sum_{v \in S \setminus P_A} I_v - \sum_{r \in \alpha_S} d_r I_r$, S is max-controlled if
$M_0^{mc}(S \setminus P_A) - \sum_{r \in \alpha_S} d_r M_0^{mc}(r) > \sum_{p \in S} I_S(p)(max_{p^\bullet} - 1)$.

For example, $S_6 = \{p_4, p_7, p_{11}, v_1, v_2\}$ is hybrid with $\bar{S_6} = p_2 + p_3 + 2p_6$. We have $\alpha_{S_6} = \{p_{10}, v_3\}$ with $d_{p_{10}} = 1$ and $d_{v_3} = 1$. As a result, S_6 is max-controlled if $M_0^{mc}(p_{11}) + M_0^{mc}(v_1) + M_0^{mc}(v_2) - M_0^{mc}(p_{10}) - M_0^{mc}(v_3) > 1$.

Algorithm 3.3

controllability constraint generation for siphons in (N^{mc}, M_0^{mc})

Input : a set of uncontrolled siphons Π_u

Output : a set of inequality constraints C

begin{

divide Π_u into Π_G, Π_H, and Π_V

derive constraints for siphons in Π_G (Π_H ; Π_V) by Propositions 3.2 (3.3 ; 3.4)

denote the set of constraints for siphons in Π_G *(*Π_H *;* Π_V*) by* C_G *(*C_H*;*C_V*) and let* $C := C_G \cup C_H \cup C_V$

Reorder the variables in each constraint in C such that no resource place is in the left side and no monitor is in the right side

output C

}end of the algorithm

Let $C = \{c_i | i \in \mathbb{N}_{|\Pi_u|}\}$ be the set of constraints of siphons in a net with $P_V = \{v_1, v_2, \ldots, v_m\}$ and $P_R = \{r_1, r_2, \ldots, r_k\}$. A constraint $c_i \in C$ can be represented by

$$c_i \equiv \sum_{j=1}^{m} \beta_j^{S_i} M_0^{mc}(v_j) < \sum_{j=1}^{k} \delta_{r_j}^{S_i} M_0^{mc}(r_j) - \omega_{S_i} \tag{3.1}$$

where $\beta_j^{S_i}$ and $\delta_{r_j}^{S_i}$ are integers and $\omega_{S_i} = \sum_{p \in S_i} I_{S_i}(p)(max_{p^\bullet} - 1)$.

Eq. (3.1) can be re-written as

$$c_i \equiv \sum_{j=1}^{m} \beta_j^{S_i} M_0^{mc}(v_j) \leq \sum_{j=1}^{k} \delta_{r_j}^{S_i} M_0^{mc}(r_j) - \omega_{S_i} - 1 \tag{3.2}$$

Let $L = [l_{ij}]_{n \times m} = [\beta_j^{S_i}]_{n \times m}$, $\mathbf{x} = [M_0^{mc}(v_1), M_0^{mc}(v_2), \ldots, M_0^{mc}(v_m)]^T$, and $\overline{B} = [\overline{b}_{ij}]_{n \times k} = [\delta_{r_j}^{S_i}]_{n \times k}$, $\omega = [-\omega_{S_1} - 1, -\omega_{S_2} - 1, \ldots, -\omega_{S_n} - 1]^T$, $B = [\overline{B}|\omega]$, and $\mathbf{y} = [M_0^{mc}(r_1), M_0^{mc}(r_2), \ldots, M_0^{mc}(r_k), 1]^T$. The controllability constraints of uncontrolled siphons can be written to be

$$L\mathbf{x} \leq B\mathbf{y} \tag{3.3}$$

Eq. (3.3) is called a (liveness) constraint equation. To find a controlled system given a plant model, the marking of monitors can be decided by solving the following LPP :

$$z = max\{\sum_{i=1}^{m} M_0^{mc}(v_i)\}$$

s.t.

$$L\mathbf{x} \leq B\mathbf{y}$$

Take the net shown in Fig. 3.2(b) as an example. The constraints of uncontrolled siphons are as follows :

$$\begin{cases} S_1 : M_0^{mc}(v_1) \leq M_0^{mc}(p_9) + M_0^{mc}(p_{10}) - 2 \\ S_2 : M_0^{mc}(v_2) \leq M_0^{mc}(p_{10}) + M_0^{mc}(p_{11}) - 1 \\ S_3 : M_0^{mc}(v_1) + M_0^{mc}(v_2) \leq M_0^{mc}(p_9) + M_0^{mc}(p_{10}) + M_0^{mc}(p_{11}) - 2 \\ S_4 : -M_0^{mc}(v_1) - M_0^{mc}(v_2) \leq -M_0^{mc}(p_9) - M_0^{mc}(p_{11}) - 2 \\ S_5 : M_0^{mc}(v_3) - M_0^{mc}(v_2) \leq M_0^{mc}(p_9) - 3 \\ S_6 : M_0^{mc}(v_3) - M_0^{mc}(v_1) - M_0^{mc}(v_2) \leq M_0^{mc}(p_{11}) - M_0^{mc}(p_{10}) - 2 \\ S_7 : M_0^{mc}(v_3) - M_0^{mc}(v_2) \leq M_0^{mc}(p_9) + M_0^{mc}(p_{11}) - M_0^{mc}(p_{10}) - 3 \end{cases}$$

The matrix form of the controllability constraints is as follows :

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ -1 & -1 & 0 \\ 0 & -1 & 1 \\ -1 & -1 & 1 \\ 0 & -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} M_0^{mc}(v_1) \\ M_0^{mc}(v_2) \\ M_0^{mc}(v_3) \end{pmatrix} \leq \begin{pmatrix} 1 & 1 & 0 & -2 \\ 0 & 1 & 1 & -1 \\ 1 & 1 & 1 & -2 \\ -1 & 0 & -1 & -2 \\ 1 & 0 & 0 & -3 \\ 0 & -1 & 1 & -2 \\ 1 & -1 & 1 & -3 \end{pmatrix} \cdot \begin{pmatrix} M_0^{mc}(p_9) \\ M_0^{mc}(p_{10}) \\ M_0^{mc}(p_{11}) \\ \mu_0 \end{pmatrix} \tag{3.4}$$

We suppose $M_0 = h_1p_1 + h_5p_5 + 5p_9 + 4p_{10} + 3p_{11}$ is an initial marking for the net in Fig. 3.2(a), where h_1 and h_5 are integers and big enough such that no transition is disabled due to deficiency of tokens in idle places. At initial marking $M_0 = h_1p_1 + h_5p_5 + 5p_9 + 4p_{10} + 3p_{11}$, the above constraints can be re-written as

$$\begin{cases} M_0^{mc}(v_1) \leq 7 \\ M_0^{mc}(v_2) \leq 6 \\ M_0^{mc}(v_1) + M_0^{mc}(v_2) \leq 10 \\ -M_0^{mc}(v_1) - M_0^{mc}(v_2) \leq -10 \\ M_0^{mc}(v_3) - M_0^{mc}(v_2) \leq 2 \\ M_0^{mc}(v_3) - M_0^{mc}(v_1) - M_0^{mc}(v_2) \leq -3 \\ M_0^{mc}(v_3) - M_0^{mc}(v_2) \leq 1 \end{cases}$$

Taking $z = max\{\sum_{i=1}^{3} M_0^{mc}(v_i)\}$ as an objective function, the LPP with the above constraints has an optimal solution $z^* = 17$ with $M_0^{mc}(v_1) = 4$, $M_0^{mc}(v_2) = 6$, and $M_0^{mc}(v_3) = 7$. That is to say, the net structure shown in Fig. 3.2(b) at initial marking $M_0 = h_1p_1 + h_5p_5 + 5p_9 + 4p_{10} + 3p_{11} + 4v_1 + 6v_2 + 7v_3$ is a controlled system with liveness for the net shown in Fig. 3.2(a) at the given initial marking M_0. Eq.(3.4) shows the liveness requirements for the net in Fig. 3.2(a). Next section focuses on reducing the size of a constraint equation.

3.4 Redundancy Identification of Constraints

Due to a large number of siphons in a controlled system, the size of a constraint equation is in theory exponential with respect to the structural scale of a Petri net model. In the standard form of an LPP, i.e, $Ax \leq b$, constraint matrix A is usually assumed to be of full row rank, where b is right-hand-side vector. It is easy to see that in this study, matrix L, in a general case, does not have full row rank, which can be shown by the existing examples in the literature. On the other hand, a designer always hopes to have a concise constraint equation. This section focuses on identifying redundant ones in order to find a small set of siphon controllability conditions.

Let (N, M_0) be a plant net model with minimal initial marking M_0^m and (N^{mc}, M_0^{mc}) be a controlled system for (N, M_0^m). The set of siphons to be controlled in (N^{mc}, M_0^{mc}) is denoted by Π_u. As stated in the previous section, the controllability of siphons can be represented by a set of constraints C, taking the form of $L\mathbf{x} \leq B\mathbf{y}$. A single constraint is denoted by (l_i, b_i) that represents $l_i^T\mathbf{x} \leq b_i^T\mathbf{y}$. Alternatively, a constraint $c_i \in C$ can be written as $(l_i, b_i) \equiv l_i^T\mathbf{x} \leq b_i^T\mathbf{y}$.

Definition 3.7 A constraint (l, b) is said to be redundant with respect to (l_i, b_i) if the truth of (l_i, b_i) implies that of (l, b).

Definition 3.8 A constraint (l, b) is said to be redundant with respect to a set of constraints C_S if the truth of one or more constraints in C_S implies that of (l, b).

Definition 3.9 Let l_α, l_β, ..., and l_γ ($\{\alpha, \beta, \ldots, \gamma\} \subseteq \mathbb{N}_{|\Pi_u|}$) be a linearly independent maximal set of matrix L. Then $C_E = \{(l_\alpha, b_\alpha), (l_\beta, b_\beta), \ldots, (l_\gamma, b_\gamma)\}$ is called a set of elementary constraints in C.

Definition 3.10 $(l, b) \notin C_E$ is called a strongly dependent constraint if $\exists a_i \geq 0$ and $(l_i, b_i) \in C_E$ such that $l = \sum_{(l_i,b_i)\in C_E} a_i l_i$.

Definition 3.11 $(l, b) \notin C_E$ is called a weakly dependent constraint if $\exists a_i > 0$ and $\exists$ non-empty $A, B \subset C_E$ such that $A \cap B = \emptyset$ and $l = \sum_{(l_i,b_i)\in A} a_i l_i - \sum_{(l_i,b_i)\in B} a_i l_i$.

Theorem 3.1

$|C_E| = rank(L)$, where $rank(L)$ denotes the rank of L.

Proof : It follows immediately from the definition of elementary constraints. □

Since the rank of L is bounded by the smaller of $|P_V|$ and $|\Pi_u|$, Theorem 3.1 leads to the following important conclusion.

Theorem 3.2

$|C_E| \leq |P_V|$.

Proof : $|C_E| = rank(L) \leq min\{|P_V|, |\Pi_u|\} \leq |P_V|$. □

Theorem 3.3

A strongly dependent constraint (l, b) is redundant with respect to C_E if $\sum_{(l_i,b_i)\in C_E} a_i b_i^T \mathbf{y} \leq b^T \mathbf{y}$.

Proof : It immediately follows due to $l^T \mathbf{x} = \sum_{(l_i,b_i)\in C_E} a_i l_i^T \mathbf{x} \leq \sum_{(l_i,b_i)\in C_E} a_i b_i^T \mathbf{y} \leq b^T \mathbf{y}$ that results from $l = \sum_{(l_i,b_i)\in C_E} a_i l_i$. □

Theorem 3.4

A weakly dependent constraint (l, b) is redundant with respect to C_E if
$\sum_{(l_i,b_i)\in C_E} a_i b_i^T \mathbf{y} - \sum_{(l_j,b_j)\in C_E} a_j b_j^T \mathbf{y} \leq b^T \mathbf{y}$.

Proof : This result is trivially true. □

For example, $C_E = \{(l_1, b_1), (l_2, b_2), (l_6, b_6)\}$ is a set of elementary constraints of Eq. (3.4). We have $l_3 = l_1 + l_2$, $l_4 = -l_1 - l_2$, $l_5 = l_1 + l_6$, and $l_7 = l_1 + l_6$. By Definitions 3.10 and 3.11, (l_3, b_3), (l_5, b_5), and (l_7, b_7) are strongly dependent and (l_4, b_4) is weakly dependent. Specifically, the redundancy condition of (l_3, b_3) is

$$M_0^{mc}(p_9) + M_0^{mc}(p_{10}) - 2 + M_0^{mc}(p_{10}) + M_0^{mc}(p_{11}) - 1 \leq M_0^{mc}(p_9) + M_0^{mc}(p_{10}) + M_0^{mc}(p_{11}) - 2,$$

i.e., $M_0^{mc}(p_{10}) \leq 1$. The redundancy conditions of the dependent constraints in Eq. (3.4) are summarized in Table 3.1.

Tab. 3.1 – Redundancy conditions of the dependent constraints in Fig. 3.2(b)

dependent constraint	redundancy condition
$l_3 = l_1 + l_2$	$M_0^{mc}(p_{10}) \leq 1$
$l_4 = -l_1 - l_2$	$M_0^{mc}(p_{10}) \geq 2.5$
$l_5 = l_1 + l_6$	$M_0^{mc}(p_{11}) \leq 2$
$l_7 = l_1 + l_6$	$M_0^{mc}(p_{10}) \leq 2$

3.5 Deadlock Prevention Policy

This section proposes a deadlock prevention policy based on the results obtained in the previous sections. Its computational complexity is also discussed.

Let Π_B denote the set of minimal siphons that are also traps marked at the minimal initial marking of an M-net. They are called B-siphons. In fact, Π_B can be easily computed from (N^{mc}, M_0^{mc}) by its structural properties. For example, $\{p_3, p_5\}$, $\{p_2, p_4, p_6\}$, and $\{p_1, p_2, p_3, p_4\}$ are B-siphons since they are marked at any admissible initial marking. By using the MIP-based deadlock detection method [16] or complete siphon enumeration [23], it can be verified whether a plant M-net contains deadlocks. We assume that it has deadlocks.

Algorithm 3.4

controlled system design for (N, M_0)

Input : an M-net (N, M_0) *with* $N = (P^0 \cup P_A \cup P_R, T, F, W)$

Output : controlled system (N^c, M_0^c)

begin{

find (N^m, M_0^m) *for* (N, M_0), *where* $N^m = N$

design a controlled system (N^{mc}, M_0^{mc}) *for* (N^m, M_0^m) *by Algorithm 3.2*

compute the set Π_A *of all minimal siphons in* N^{mc}

find Π_B

$\Pi_u := \Pi_A \setminus \Pi_B$

derive controllability constraints for siphons in Π_u *by Algorithm 3.3*

find redundancy conditions for constraints

decide $M_0^c(v)$ *by reduced constraint set and* M_0, $\forall v \in P_V$

$\forall p \in P^0 \cup P_A \cup P_R$, $M_0^c(p) := M_0(p)$

$N^c := N^{mc}$

output (N^c, M_0^c)

}end of the algorithm

Theorem 3.5

(N^c, M_0^c) *resulting from Algorithm 3.4 is a live controlled system for plant model* (N, M_0).

Proof : All siphons in (N^c, M_0^c) are max-controlled, implying that it satisfies cs-property. From

Definition 3.1, (N^c, M_0^c) is live. □

The complexity of this deadlock control algorithm is exponential with respect to the size of a net model since both the theory of regions and a complete siphon enumeration are exponential. However, the fact underlying this policy is that, given a plant model with any admissible initial marking, its controlled system can be easily decided by the controllability constraints of siphons once the structure of a controlled system is determined. That is to say, given a plant model, we just need to use the theory of regions and compute all minimal siphons once to find the structure of a controlled system. Even if the initial marking of the plant model changes, the structure of the controlled system obtained previously can be reused. This means that we just need to find the markings of the monitors in the new controlled system, which can be decided by satisfying the controllability constraints of siphons. Fig. 3.3 shows the flowchart underlying the deadlock control strategy, where the computation involved in the steps above the dotted line is carried out only once for a net N.

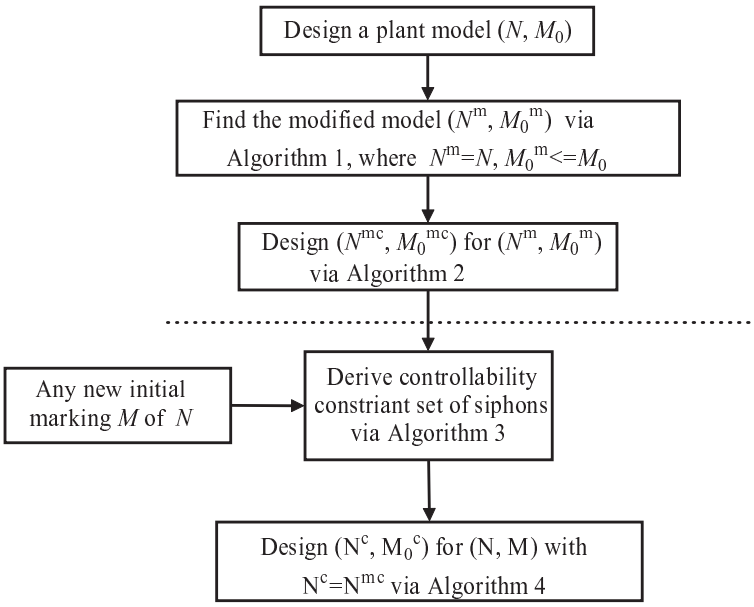


FIG. 3.3 – Flowchart of the deadlock prevention policy.

One may wonder the performance of the proposed method since many approaches in the literature need to compute siphons only once given a net structure, e.g., those in [23], [69], [46], [47], and [48]. However, they are usually overly conservative. For example, a study on a typical AMS shows that the controlled system due to [23] usually has 30% or so permissive behavior of

an optimal one [74]. The next section discusses the performance of the proposed method.

3.6 Examples

This section considers two typical examples that are widely investigated in the literature, indicating that the proposed deadlock prevention policy is nearly optimal.

An AMS consists of two robots R1 and R2 and three machines M1–M3. Its model is shown in Fig. 3.4(a). It is an M-net, where p_1 and p_{10} are idle places, $p_{11} - p_{15}$ are resource places, and the others are activity places. (N^{mc}, M_0^{mc}) shown in Fig. 3.4(b) is the controlled system for the plant net with the minimal initial marking. Seven uncontrolled siphons in (N^{mc}, M_0^{mc}) are $S_1 = \{p_6, p_8, p_{13}, p_{14}\}$, $S_2 = \{p_5, p_8, v_2, v_3\}$, $S_3 = \{p_5, p_7, p_{12}, p_{13}\}$, $S_4 = \{p_6, p_7, p_{12}, p_{14}, v_2, v_3\}$, $S_5 = \{p_6, p_8, p_{14}, v_2, v_3\}$, $S_6 = \{p_6, p_7, p_{12}, p_{13}, p_{14}\}$, and $S_7 = \{p_5, p_7, p_{12}, v_2, v_3\}$. The matrix form of the controllability constraints is as follows :

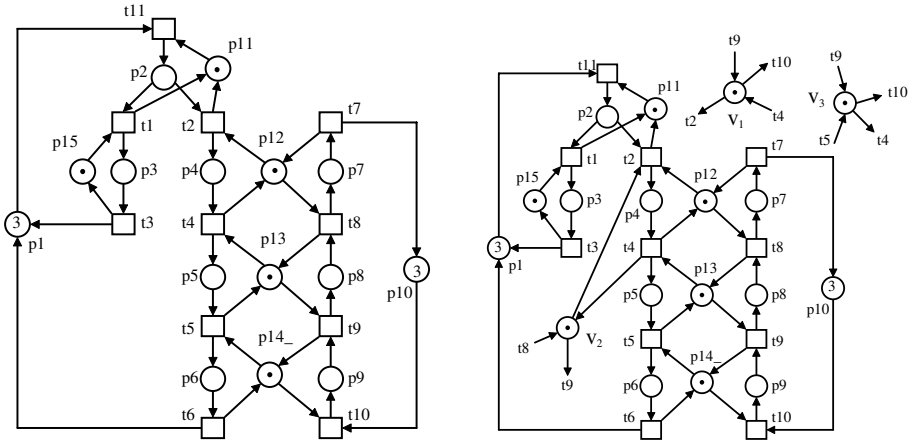


FIG. 3.4 – (a) An M-net (N, M_0), (b) controlled system (N^{mc}, M_0^{mc}).

$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & -1 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} M_0^{mc}(v_1) \\ M_0^{mc}(v_2) \\ M_0^{mc}(v_3) \end{pmatrix} \leq \begin{pmatrix} 0 & 1 & 1 & -1 \\ 0 & 0 & 0 & -1 \\ 1 & 1 & 0 & -1 \\ 1 & 0 & 1 & -1 \\ 0 & 0 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ 1 & 0 & 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} M_0^{mc}(p_{12}) \\ M_0^{mc}(p_{13}) \\ M_0^{mc}(p_{14}) \\ 1 \end{pmatrix} \tag{3.5}$$

Table 3.2 shows the permissive behavior of controlled systems at different initial markings,

where the initial markings of the monitors are decided by Eq. (3.5). In this table, B_p is the number of reachable states of (N, M_0), B_L represents the number of states that an optimal controlled system for (N, M_0) has, B_m indicates the number of states of controlled system (N^{mc}, M_0^{mc}), and B_m/B_L implies the optimality degree.

Tab. 3.2 – Behavioral permissiveness of the proposed deadlock prevention policy in Fig. 3.4(b)

$p_1, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}, p_{15}$	v_1, v_2, v_3	B_p	B_L	B_m	B_m/B_L
3, 3, 1, 1, 1, 1, 1	1, 1, 1	73	54	54	100%
4, 4, 2, 2, 2, 2, 2	3, 3, 2	1093	1047	941	89.88%
5, 5, 3, 3, 3, 3, 3	5, 5, 3	5767	5705	5151	90.29%
6, 6, 4, 4, 4, 4, 4	7, 7, 4	20324	20263	18517	91.38%
7, 7, 5, 5, 5, 5, 5	9, 9, 5	57450	57390	52995	92.25%
8, 8, 6, 6, 6, 6, 6	11, 11, 6	140703	140643	131000	93.14%
9, 9, 7, 7, 7, 7, 7	13, 13, 7	310783	310723	291363	93.77%
10, 10, 8, 8, 8, 8, 8	15, 15, 8	634173	634113	597853	94.28%
11, 11, 9, 9, 9, 9, 9	17, 17, 9	1214679	1214619	1150189	94.70%
12, 12, 10, 10, 10, 10, 10	19, 19, 10	2208445	2208385	2098887	95.04%

The second AMS is shown in Fig. 3.5(a). It has two robots R1 and R2, each of which can hold one product at a time. The cell also contains four machines M1–M4, and each of them can hold one part. Parts enter AMS through two automatic loading buffers I1 and I2, and leave it through two unloading ones O1 and O2. The robots deal with the movements of parts. Two part types P1 and P2 are produced. Their respective production routes are shown in Fig. 3.5(b).

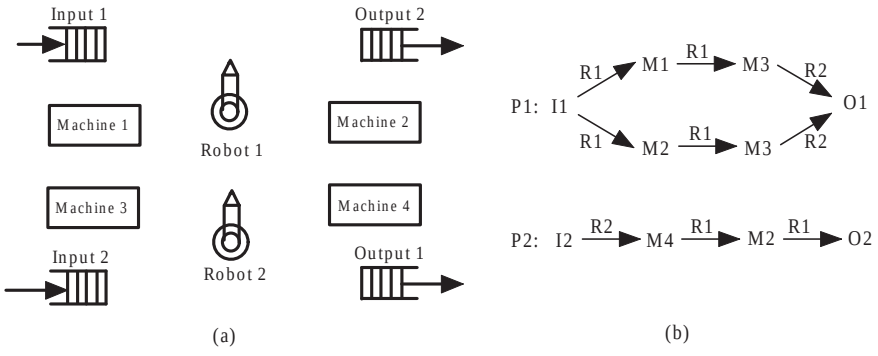


Fig. 3.5 – (a) Layout of an AMS, (b) routes of part types P1 and P2.

Fig. 3.6(a) shows its net model, which is an M-net with $P^0 = \{p_1, p_8\}$, $P_R = \{p_{15}, p_{16}, p_{17}, p_{18}, p_{19}\}$, and the others are activity places. It can be easily verified that the current initial marking is

minimal. The controlled system of such a plant model is depicted in Fig. 3.6(b), which can be obtained by the theory of regions [104]. In Fig. 3.6(b), there are 54 uncontrolled minimal siphons. For economy of space, the liveness constraint equation for the system is not shown. Table 3.3 shows the performance of the controlled systems at different initial markings. From this table, we conclude that the proposed method for this example is near-optimal.

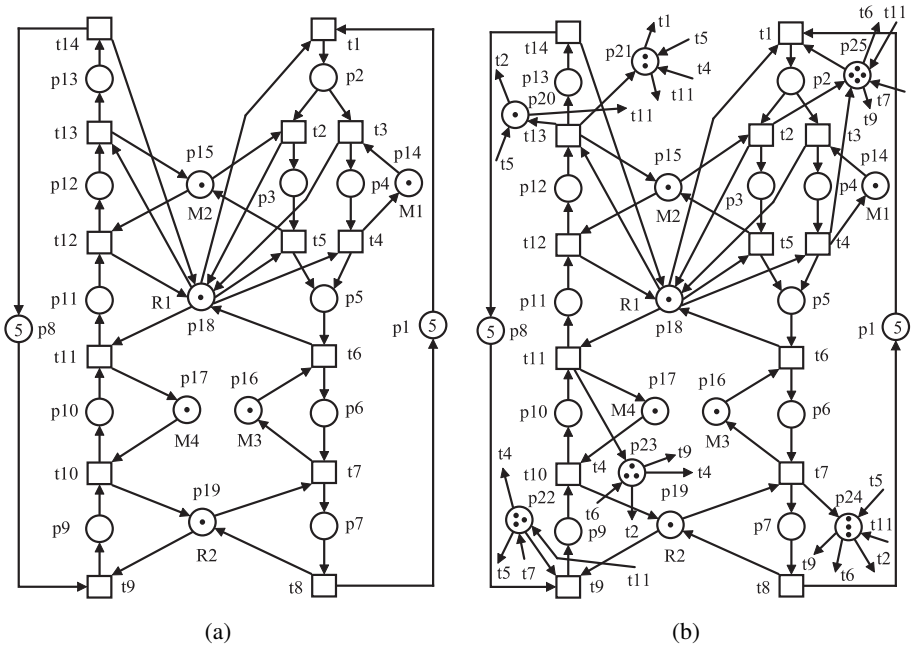


FIG. 3.6 – (a) Petri net model of an AMS, (b) structure of the controlled system.

TAB. 3.3 – Behavioral permissiveness of the proposed deadlock prevention policy in Fig. 3.6(b)

$p_1, p_8, p_{14} - p_{19}$	$p_{20} - p_{25}$	B_p	B_L	B_m	B_m/B_L
3, 3, 1, 3, 1, 2, 3, 2	5, 4, 5, 8, 7, 7	2946	2945	2842	96.50%
3, 4, 1, 4, 1, 1, 3, 3	4, 4, 7, 9, 8, 9	5235	5233	4730	90.35%
4, 4, 1, 5, 1, 1, 2, 4	6, 7, 6, 14, 14, 14	6877	6868	6861	99.90%
4, 5, 1, 5, 2, 2, 4, 5	5, 9, 9, 5, 10, 18	31759	31578	29129	92.24%
5, 5, 1, 6, 1, 1, 3, 5	8, 9, 7, 10, 12, 17	28243	28233	28177	99.80%
5, 6, 1, 7, 1, 6, 3, 1	9, 10, 8, 13, 13, 13	24448	24438	24384	99.78%
6, 6, 3, 7, 3, 3, 5, 7	8, 8, 10, 17, 13, 12	298725	298724	290187	97.14%

3.7 Discussions

For a generalized Petri net model, the proposed deadlock prevention policy does not provide a controlled system as permissive as the case in which N^{mc} is ordinary. From the theoretical point of view, this is due to the conservativeness of Proposition 3.1, which can be verified by the following example. Fig. 3.7(a) shows a generalized net and Fig. 3.7(b) depicts its controlled system derived from Propositions 3.1 and 3.2. For the sake of simplicity, r_1 $(r_2\,;\,v)$ is used to denote a place as well as the number of tokens in it.

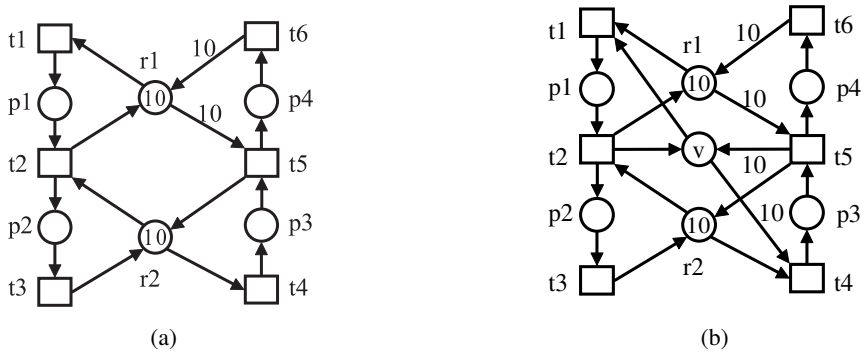


FIG. 3.7 – (a) An example net, (b) its controlled system.

The only siphon in Fig. 3.7(a) is $S = \{p_2, p_4, r_1, r_2\}$. Its controllability is ensured by the addition of monitor v, as shown in Fig. 3.7(b) [5]. By Proposition 3.2, S is max-controlled if $v < r_1 + r_2 - \sum_{p\in S} I_S(p)(max_{p^\bullet} - 1)$, where $I_S = p_2 + p_4 + r_1 + r_2 - v - 9p_3$. When $r_1 = r_2 = 10$, we have $v \leq 10$. However, v can be obtained by solving the following LPP :

$max\ v = x + y$

$x \leq r_1$

$(1/W(v, t_4))y \leq r_2$

$(r_1 + r_2) - x - (1/W(v, t_4))y \geq \sum_{p\in S} I_S(p)(max_{p^\bullet} - 1)$

This problem has an optimal solution $v = 100$ with $x = 0$ and $y = 100$. It can be verified that the net shown in Fig. 3.7(b) at initial marking $M_0 = 10r_1 + 10r_2 + 100v$ is live. Compared with $v = 10$ resulting from Propositions 3.1 and 3.2, this novel siphon control method achieves much better control effects. For the net shown in Fig. 3.2(b), when $M_0(p_9) = M_0(p_{10}) = M_0(p_{11}) = 6$, we have $M_0^{mc}(v_1) = 5$, $M_0^{mc}(v_2) = 11$, and $M_0^{mc}(v_3) = 14$ due to Eq. (3.4). However, by the novel siphon

control approach, we can have $M_0^{mc}(v_1) = 16$, $M_0^{mc}(v_2) = 11$, and $M_0^{mc}(v_3) = 16$. The net in Fig. 3.2(a) at $M_0(p_9) = M_0(p_{10}) = M_0(p_{11}) = 6$ has 12,495 reachable states. An optimal live controlled system should have 12,415 states. The controlled system at $M_0^{mc}(v_1) = 16$, $M_0^{mc}(v_2) = 11$, and $M_0^{mc}(v_3) = 16$ has 12,374 reachable states. The optimality ratio is 12,374/12,425=99.67%.

The permissiveness of a supervisor derived from the proposed deadlock prevention approach strongly depends on the control of an uncontrolled siphon. Siphon control in a generalized Petri net by Proposition 3.1 is too conservative, i.e., the addition of a monitor exclude legal states from the resulting controlled system. It is not surprising since the concept of max-controlled siphons is applicable to any net, not limiting to some special net classes. Recognizing its drawback, the concept of max′-controlled [9] and max′′-controlled siphons [80] are proposed for S^4R, which, to a large extent, relaxes the controllability condition of siphons. That is to say, more tokens can be initially marked in a monitor if a siphon is max′- or max′′-controlled. The proposed deadlock prevention policy can also lead to a nearly optimal controlled system for a generalized plant net model if the concept of max′-controlled [9] or max′′-controlled [80] siphons is, if applicable, employed to derive siphon controllability condition.

3.8 Summary

Deadlocks are a threat to the safety and performance of AMS. Deadlock prevention is a well defined strategy in resource allocation systems. It is usually developed by either structural or state space analysis. Those based on structural analysis such as siphons cannot in general lead to optimal supervisors [51], [74], while those combining state space analysis can lead to optimal or suboptimal ones [90], [91], [104], [106], [108]. For example, the theory of regions is a technique that can find an optimal one when it exists. However, its computation is notoriously expensive since a complete state enumeration is necessary and the number of LPP to be solved is exponential with respect to the size of a plant model and its initial marking.

The proposed deadlock prevention policy aims to felicitously trade off behavioral optimality for computational tractability. To achieve this, we first derive, by using the theory of regions, a supervisor for the plant model with its minimal initial marking. The controllability of siphons is expressed as a set of inequality constraints with respect to the markings of resource places and

monitors. For a fixed net structure with different initial markings, the theory of regions is used and a siphon enumeration is computed once only. Then, the supervisor can be decided by satisfying the constraints of siphon controllability by adjusting the initial marking of monitors only. Once an optimally controlled system's structure is found, a nearly optimal system can be obtained without using the theory of regions. AMS examples show that the proposed method is nearly optimal.

Future efforts will be guided to a near-optimal supervisor with low computational overhead and an efficient method to find a deadlock-liable minimal initial marking for an M-net. The performance of the proposed method for generalized net models needs to be addressed.

The major contribution in this research is published :

[1] Zhiwu Li, **Gaiyun Liu**, Hans-Michael Hanisch, and Mengchu Zhou, Deadlock prevention based on structure reuse of Petri net supervisors for flexible manufacturing systems, *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol.42, no.1, pp.178-191, 2012.

Chapitre 4

Maximally Permissive Control Policy for a Subclass of S^3PR without Reachability Analysis

4.1 Introduction

Traditional maximally permissive control policies rely on reachability analysis or MIP test and suffer from the state explosion or NP-hard problem. It has been a hot race to synthesize optimal controllers to be maximally permissive with fewest monitors. Previous work shows that among all n-dependent siphons, only one siphon (whose unmarked state follows some token distribution) needs to be controlled. This greatly simplifies the supervisor synthesis as well as minimizes the number of monitors required while making the controlled net maximally permissive (i.e., all live states can be reached). This chapter further proposes a maximally permissive control policy for a subclass of S^3PR (called β-nets) based on the above theory of token distribution pattern of unmarked siphons.

We first show that by adding a monitor for each critical siphon, some live states may get lost since the monitor controls the complementary set of a critical siphon, where some places may not be marked. By controlling only the set of marked activity places, more live states can be reached.

However, this induces some emptiable siphons. The corresponding token pattern can be inferred. By adding monitors to all such possibly emptiable siphons, the controlled net becomes live and maximally permissive. There is no need to construct reachability graph and enumerate all minimal siphons. Hence, the computational burden is the least among all approaches in the literature.

Nevertheless, the presented approach addresses a simpler problem in time complexity with less information provided compared to behavior analysis based on the reachability graph.

The rest of this chapter is organized as follows. Section 2 presents the theory of critical siphons. The control policy is developed with examples in Section 3. Several examples are used to illustrate the proposed method in Section 4. Section 5 concludes the chapter.

4.2 Critical Siphon

This section develops the theory of critical siphons. Once a critical siphon in a set Q of siphons is controlled, so will be the rest of siphons in Q. The following defines Q based on basic and compound siphons, which are synthesized from elementary and compound resource circuits c (all places in the circuit are resource types) ; respectively by attaching handles (directed paths) to c, like handles to a tea pot. An n-compound siphon S_0 is a compound siphon containing n basic siphons (serially connected). An n-dependent siphon is either an n-compound one S_0 or a siphon derived from S_0. An XY-handle starts from a node in X (='T' or 'P') to a node on Y (='T' or 'P'). For instance, A TP-handle is a directed path from a transition to a place. For more details, please refer to [96].

Definition 4.1 *An elementary resource circuit c_b in an S^3PR, i.e., all places in c_b are resources, is called a basic circuit. SMS constructed from c_b is called a basic siphon. Let S_0 be an SMS and $S_0 = S_1 \circ S_2 \circ \ldots S_{n-1} \circ S_n$ is called an n−compound siphon, where S_1, S_2, ..., and S_n are basic siphons, $S_i \cap S_j = \{r_i\}$, $r_i \in P_R$, $j = i+1$, $i = 1, 2, \ldots, n-1$. A subcompound siphon of S_0 is of the form : $S_i \circ S_{i+1} \circ \ldots S_{j-1} \circ S_j$ ($i \geq 1$ and $j \leq n$) of size $k < n$. S_0 is also called a uniform compound siphon if $\forall i$, $H(r_i) \cap S_i \cap [S_0] \neq \emptyset$. r_i is called an internal place and is said to be singular if $M_0(r_i) = 1$.*

Note that S_0 is a strongly dependent siphon by Theorem 2 in [8] since $\eta_0 = \eta_1 + \eta_2 + \cdots + \eta_{n-1} + \eta_n$ where η_0 (resp. η_i) is the characteristic T-vector of S_0 (resp. S_i). Fig. 4.1 provides an example for strongly dependent siphon. As shown in Table 4.1, $c_1 \cap c_2 = \{p_8\}$ is a single resource place and the SMS synthesized from $c_1 \cup c_2$ strongly depends on S_1 and S_2 synthesized from c_1 and c_2, respectively. The set of basic siphons equals that of elementary siphons. This is consistent

with the theory in [12], [110]. When $c_1 \cap c_2$ is no longer a single resource place but a directed path, then S_0 is no longer a strongly dependent siphon and may be a weakly one [12].

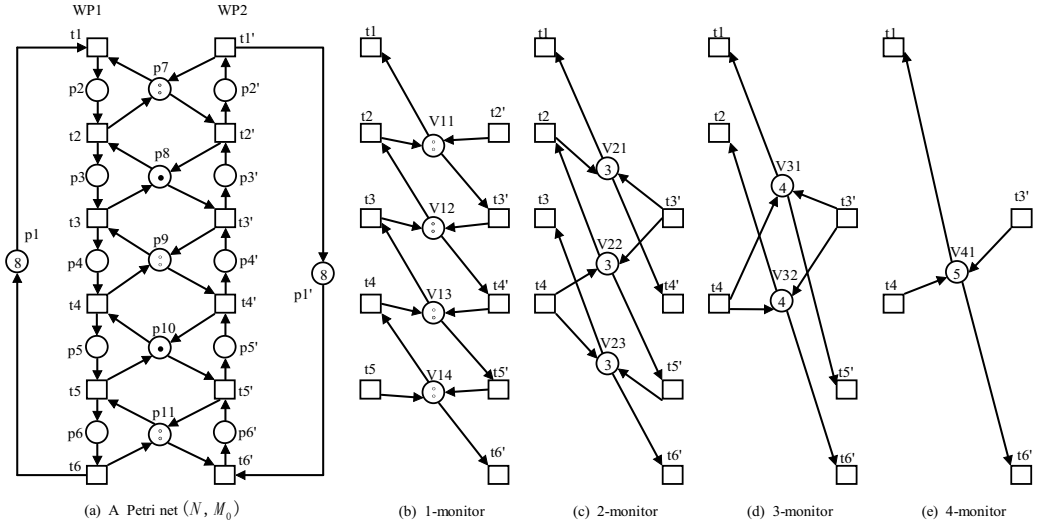


FIG. 4.1 – An example of the control policy based on UP but with no unmarked places in $H(V)$.

TAB. 4.1 – Basic siphons, resource circuits, and $V(M_0)$ for the net in Fig. 4.1(a)

S_b	c_b	V(M_0)	$c_i \cap c_{i+1}$
S_1={p_3, p_7, p_8, p'_2}	c_1=[$p_7 t'_2 p_8 t_2 p_7$]	V_{11} (2)	p_8
S_2={p_4, p_8, p_9, p'_3}	c_2=[$p_8 t'_3 p_9 t_3 p_8$]	V_{12} (2)	p_9
S_3={p_5, p_9, p_{10}, p'_4}	c_3=[$p_9 t'_4 p_{10} t_4 p_9$]	V_{13} (2)	p_{10}
S_4={$p_6, p_{10}, p_{11}, p'_5$}	c_4=[$p_{10} t'_5 p_{11} t_5 p_{10}$]	V_{14} (2)	

Definition 4.2 *$r \in P_R$ is called non-sharing if r is used by only one process. An S^3PR with no non-sharing r is called a β-net. Each siphon S^* such that $[S^*] \cap [B] \neq \emptyset$ is called a k-dependent one, where $B = S_i$ or S_j (S_i and S_j defined in Definition 4.1). The set of n-dependent siphons is denoted as £(S_0). The monitor for S^* (resp. S_i) is called a k-monitor (resp. 1-monitor).*

Note that once all basic siphons $(S_1, S_2, ..., S_n)$ are identified, so will be all subcompound ones since $S_0 = S_1 \circ S_2 \circ \ldots S_{n-1} \circ S_n$ and any compound siphon S' is of the form $S' = S_i \circ S_{i+1} \circ \ldots \circ S_{j-1} \circ S_j$, which can be easily computed using the method in [10] as all $S_i, S_{i+1}, ..., S_{j-1}$, and S_j have been found. Furthermore, we have developed theory [10] to efficiently extract SMS incrementally rather than the traditional global approach. Only linear number of basic siphons needs to be searched. Adding and deleting common sets of places from existing ones (called composition

method), one can derive the compound siphons with much reduced search time. It is easily subject to computer implementation in a very efficient way compared with all current techniques since all these steps can be expressed in terms of formulas. Only compound siphons that are critical need to be considered. Most importantly, our algorithm does not need to compute any compound siphon.

Also note that if S_0 is not uniform (denoted by 'NU'), control circuits (from which to synthesize control siphons) may not be formed from two adjacent control places V_h and V_{h+1} for S_h and S_{h+1}, respectively. As a result, there is only one siphon (i.e., S_0) in £(S_0). In most cases, S_0 is not emptiable and needs no monitors.Hence, we should focus on uniform compound siphons.

Definition 4.3 *Let $S_0 = S_1 \circ S_2 \circ \ldots S_{n-1} \circ S_n$ be a compound siphon. The token distribution (called unmarked) pattern M is as follows : (1) For each singular place r_i, $M(r_i) = 1$, $M(H(r_i) \cap [S_0]) = 0$; and (2) For other r in S_0, $M(r) = 0$, $M(H(r) \cap [S_0] \cap^\bullet (V^\bullet)) = M_0(r)$, where V is a monitor for S_0. An n-dependent siphon S with the above unmarked pattern (UP) is called a critical siphon.*

Theorem 4.1

Let M be a UP for $S_0 = S_1 \circ S_2 \circ \ldots S_{n-1} \circ S_n$, then

1. *$\forall r_j \in S_0$, $\forall t \in r_j^\bullet \cap [S_0]^\bullet$, t (any output transition of r_j) is disabled ;*
2. *$\forall r_j \in S_0$, $\forall t \in^\bullet r_j \cap^\bullet [S_0]$, t (any input transition of r_j) is disabled ;*
3. *$\exists S \in$ £(S_0), S is unmarked and $M([S]) = \sum_{r \in R(S_0)} M(r) - \theta$, where θ is the number of singular places.*
4. *Once the critical siphon S for M in Definition 4.3 is controlled, so are the rest of siphons S' with $M([S']) \neq M([S])$ in £(S_0).*

Proof :

1. There are two cases $\forall r_j \in S_0$:
   (a) $M(r_j) = 0$, t is disabled.
   (b) $M(r_j) = 1$ and $M(p) > 0$, $p \in^\bullet t$. Let V be the monitor added for $S_0^* = S_1 - S_{j+1}$, $t \in V^\bullet$, and all internal r follow the UP. Then $M(V)$=0 and t is also disabled. Similar conclusion applies when V is the monitor added for $S_0^* = S_j - S_n$ and $t \in V^\bullet$.
2. There are two cases $\forall r_j \in S_0$:

(a) r_j is internal. Then t is also an output transition of an internal resource place. By Part 1 of this theorem, t is disabled.

(b) $r = r_1, r_{n+1}$. t is disabled since $M(p) = 0, p \in^{\bullet} t \cap S_0$.

3. All input and output transitions of the monitor are dead, they remain so permanently. The set of unmarked places form an unmarked siphon from which we can extract a minimal siphon S such that $[S] \cap [B] \neq \emptyset$, where $B = S_1$ or S_n. By Definition 4.3, $S \in \pounds(S_0)$. $M([S]) = \sum_{r \in R(S_0)} M(r) - \theta$ by Definition 4.3 since M is a UP by assumption.

4. There are two cases :

(a) $M_{max}([S']) < M_{max}([S])$. By the theory in [96], $[S_0] \supset [S_c] \supset [S_x^p] \supset [S_x^f]$ where S_c, S_x^p, S_x^f are control, partial mixture, and full mixture siphons, respectively. Furthermore, $S_c \cap [S_x^p] \neq \emptyset$ and $S_x^p \cap [S_x^f] \neq \emptyset$. Thus, $[S'] \subset [S]$, $S' \cap [S] \neq \emptyset$, and S' is marked at the UP M. Adding the monitor for S will push tokens out of $[S]$ to make some control places in S' marked. Hence S' is always marked or controlled.

(b) $M_{max}([S']) > M_{max}([S])$. This is impossible since all output and input transitions of S_0 are dead at M, where $M([S]) = M_{max}([S])$. One can no longer fire any transition to move tokens into $[S'] \subset [S_0]$ to make $M_{max}([S']) > M_{max}([S])$. After adding V, $M([S]) < M_{max}([S])$ and transitions are disabled more easily to make S' more marked (i.e., have more tokens). Thus, S' is controlled.

□

In Fig. 4.1(a), there are four basic circuits from which, four basic siphons $S_1 = \{p_3, p_7, p_8, p_2'\}$, $S_2 = \{p_4, p_8, p_9, p_3'\}$, $S_3 = \{p_5, p_9, p_{10}, p_4'\}$, $S_4 = \{p_6, p_{10}, p_{11}, p_5'\}$ can be synthesized, and four monitors V_{11}, V_{12}, V_{13}, and V_{14} in Fig. 4.1(b) are added accordingly. For 2- and 3- dependent siphons, the UP are obvious and monitors are added as shown in Figs. 4.1(c) and (d), respectively. For the only 4-dependent siphon S, p_8, p_9, and p_{10} are internal places ; p_8 and p_{10} are singular. $\theta = 2$ and the UP (see Fig. 4.1(e) for the monitor) is $M = 2p_2 + 2(p_4 \oplus p_4') + 2p_6' + p_8 + p_{10}$, where $2(p_4 \oplus p_4')$ indicates that $M(p_4) + M(p_4') = 2$.

The controlled model reaches 1060 states losing 66 live states (there are 1126 live states among all 1432 reachable states) by adding a monitor V to each critical siphon with $H(V) = [V] = [S] = \{p_2, p_3, p_4, p_4', p_5', p_6'\}$ (unmarked p_3 and p_5' are not excluded since the presence of unmarked

places clutters the figure.), where $H(V)$ is the set of holder places that use V. By shifting a token from p_2 to p_3, M is changed to $M' = p_2 + p_3 + 2(p_4 \oplus p'_4) + 2p'_6 + p_8 + p_{10}$, which is a live marking since $p_8 \in S$ or $p'_2 \in S$ holds a token. But M' is forbidden by the monitor added for S since $M([S]) = M'([S])$. This can be avoided by shrinking $[S]$ so that unmarked activity places in $[S]$ are precluded from being controlled as shown below.

Definition 4.4 *Let $\{p_1, p_2, p_3\} \subseteq [S] \cap P_{Ai}$ be a set of 3 consecutive activity places in subprocess N_i. $p_j \in p^{\bullet\bullet}_{j-1}$, $j = 2, 3$. p_2 is said to be a hole in $[S]$ at M if only $M(p_2) = 0$.*

To reach more states for the net shown in Fig. 4.1(a), one should set $H(V) = \Psi = [S]\backslash\{p_3, p'_5\}$, the set of marked activity places or UP. Note that $M_{max}(\Psi) = M_0(R(\Psi))$ and one cannot shift a token from p_2 to p_3 to forbid live states that happens when $H(V) = [S]$ where $M_{max}([S]) < M_0(R([S])$.) However, this induces new emptiable siphons and the net is not live implying that there are other token distributions (e.g., $M' = 2p_2 + p_3 + p'_4 + 2p'_6 + p_9 + p_{10}$) which corresponds to another forbidden marking rather than the unmarked one. Note that M' is a forbidden (necessarily evolves to M by firing t_3) but not a dead (since t_3 is enabled) marking in the uncontrolled net. It is a dead one after V is added to prevent M from being reached by setting $H(V) = \Psi$ and all k-dependent ($k < 4$) siphons have been controlled.

The only possible enabled transition is $t \in p^{\bullet}_9$. Let $t = t_3 \in p^{\bullet}_9 \cap p^{\bullet}_3$. Then t_3 is also an output transition of V; i.e., $t_3 \in V^{\bullet}$. $M'(V) = 0$ and t_3 is disabled at M'. There is another output transition $t' = t'_4 \in {}^{\bullet}p'_4$ of $r' = p_9$. Let V' be the monitor for a critical siphon S' ($S_0 = S_1 \circ S_2$) such that $r, r' \in S'_R = \{p_7, p_8, p_9\}$ and $t'_4 \in V'^{\bullet}$. Then $M'(V') = 0$ and t' is also disabled at M'. Thus, all transitions in the net are disabled at M' after V is added. Hence, M' is a dead marking.

M' is obtained from M by shifting a token from p_8 to p_9 (or from p_4 to p_3); this operation is called *exchange* one below. There are only these two kinds of forbidden markings (the above forbidden ones and the unmarked pattern ones in Theorem 4.1).

Definition 4.5 *Let S be an emptiable n-dependent siphon, $\Psi \subset [S]$ a set of marked activity places when $M(S) = 0$, $M \in R(N, M_0)$. The operation of shifting a token from a place $p \in H(r)$ in Ψ at M to another place p' in $[S]$ to form another marking $M' \in R(N, M_0)$ is called an exchange operation. M' is called an exchanged unmarked pattern (EUP). The internal place r' such that $p' \in H(r')$ is called a boundary place.*

Note that both p and p' must be in $[S]$ since otherwise the set of unmarked activity places Ψ' at M' may correspond to a different siphon. In the sequel, all M' encountered denotes the above M' in Definition 4.5 and M'_Ψ denotes the submarking obtained by projecting M' onto Ψ.

Note that $M^* = 2p_2+p_3+p_4+2p'_6+p_9+p_{10} = M-2(p_4 \oplus p'_4)+p_3+((M_0(p_9)-1))p_4$ can also be obtained by the same exchange operation ; however, M^* is a live marking ($M^*(p_9) = 1, t'_4 \in {}^\bullet p'_4$). We need to find the exact token pattern after the exchange operation to find the corresponding set Ψ' of marked activity places to add correct control arcs.

Theorem 4.2

Let S, Ψ, M, M', $p \in H(r)$ and $p' \in H(r')$ be as defined in Definition 4.5. $H(r) = \{p, p^+\}$. V is the monitor added to control S and $H(V) = \Psi$ (to reach more states). All k-monitors ($k < n$) have been added.

1. *M''=$M + p' + (M_0(r) - 1)p^+ - M_0(r)(p^+ \oplus p)$ is a forbidden marking.*
2. *$M^\star$=$M + p' + ap + bp^+ - M_0(r)(p^+ \oplus p)$ is a live marking, $a + b = M_0(r) - 1, a > 0$.*

Proof :

1. The only possible enabled transition is $t \in r^\bullet$. Let $t \in r^\bullet \cap p^\bullet$. Then t is also an output transition of V ; i.e., $t \in V^\bullet$. $M''(V) = 0$ and t is disabled at M''. There is another output transition $t' \in {}^\bullet p^+$ of r. Let V' be the monitor for a critical siphon S' such that $r, r' \in S'_R$ and $t \in V'^\bullet$. Then $M''(V') = 0$ and t' is also disabled at M''. Thus, all transitions in the net are disabled at M'' after V is added. Hence, M'' is a forbidden marking.
2. At $M^\star$, $M^\star(V') > 0$ ($M^\star(S') > 1$) and t' is enabled. After t' fires, other transitions become enabled and so on. The marking $M^\star$ is a live one.

□

Note that we cannot move a token from p_2 (on the opposite side of p_3 in contrast to p_4) to p_3 since then the 4-dependent siphon becomes marked. We cannot move a token from p_{10} to p_9 (or from p_4 to p_5) since then S_4 becomes unmarked which is impossible as S_4 (4^{th} basic siphon) has been controlled by adding a monitor.

4.3 Control Policy

This section develops a maximally permissive control policy based on the token patterns for unmarked siphons.

To reach more states, all possible sets of unmarked activity places must be considered to add a monitor accordingly. They can be identified by finding all possible exchange operations. For the above siphons, such a policy results in a maximally permissive controlled net.

Now we give a detailed picture of the controlled model in Fig. 4.1. The only available exchange operations are with 3- and 4-dependent siphons. Consider $S_0 = S_1 \circ S_2 \circ S_3$. The UP is $M_\Psi = 2p_2 + 2(p_4 \oplus p'_4) + p'_5$. By shifting a token from p_4 to p_3, we have $M'_{\Psi'} = 2p_2 + p_3 + p'_4 + p'_5$. Adding a monitor V to each of Ψ and Ψ' ($H(V) = \Psi$ and Ψ') with $M_0(V) = M_0(S) - \theta - 1 = 6 - 2 = 4$, the resulting model now reaches $1074 > 1060$ states.

Similar discussion applies to $S'_0 = S_2 \circ S_3 \circ S_4$. Unmarked pattern is $M^*_{\Psi^*} = p_3 + 2(p_4 \oplus p'_4) + 2p'_6$. Adding a monitor V^* with $[V^*] = \Psi^*$ induces a new siphon, whose UP can be reached by (exchange operation) moving a token from p'_4 to p'_5 (associated with singular place p_{10}) such that $M^*_{\Psi^\wedge} = p_3 + p_4 + p'_5 + 2p'_6$ that is another forbidden state besides $M^*_{\Psi^*}$. By adding a monitor for each of Ψ^* and $\Psi^\wedge$, the resulting model reaches 1082 states.

Finally, consider the only 4-monitor with singular places p_8 and p_{10}. With 2 possible Ψ's for each singular place, there are four possible sets of unmarked activity places. Hence, we replace the monitor with 4 monitors. The resulting model as shown in Table 4.2 reaches 1126 states which is maximally permissive. Note that when $M_0(p_9) = 1$, all internal places are singular and all critical siphons S are control ones. The marking for each unmarked S is a UP.

The controller synthesis is summarized in Algorithm 4.1.

Algorithm 4.1

Controller Synthesis for β-nets

INPUT : An uncontrolled β-net

OUTPUT : Maximally permissive controlled system

1. For each basic siphon, add a monitor V with $M_0(V) = M_0(S)$-1 ;

2. For each n-dependent region ($n \geq 2$) Θ,

(2.1) For each non-boundary place in Θ, follow Definition 4.3 to find its token distribution ;

(2.2) For each boundary place in Θ, find all possible exchange operations and corresponding token patterns ;

(2.3) Combine token patterns obtained in Steps 2.1 and 2.2. For each such a token pattern, add a monitor with $M_0(V) = M_0(S) - \theta - 1$, where θ is the number of singular places in S ; Output the resulting controlled model.

Tab. 4.2 – Controlled model for the net in Fig. 4.1.

S	$V(M_0)$	$V_S^\bullet$	${}^\bullet V_S$	$[V_S]$
S_1	$V_{11}(2)$	t_1, t'_3	t_2, t'_2	p_2, p'_3
S_2	$V_{12}(2)$	t_2, t'_4	t_3, t'_3	p_3, p'_4
S_3	$V_{13}(2)$	t_3, t'_5	t_4, t'_4	p_4, p'_5
S_4	$V_{14}(2)$	t_4, t'_6	t_5, t'_5	p_5, p'_6
S_5	$V_{21}(3)$	t_1, t'_4	t_2, t'_3	p_2, p'_4
S_6	$V_{22}(3)$	t_2, t'_5	t_4, t'_3	p_3, p_4, p'_4, p'_5
S_7	$V_{23}(3)$	t_3, t'_6	t_4, t'_5	p_4, p'_6
S_8^1	$V_{31}^1(4)$	t_1, t'_5	t_3, t'_3	p_2,p_3,p'_4,p'_5
S_8^2	$V_{31}^2(4)$	t_1, t_3, t'_5	t_2, t_4, t'_3	p_2, p_4, p'_4, p'_5
S_9^1	$V_{32}^1(4)$	t_2, t'_6	t_4, t'_4	p_3, p_4, p'_5, p'_6
S_9^2	$V_{32}^2(4)$	t_2, t'_4, t'_6	t_4, t'_3, t'_5	p_3, p_4, p'_4, p'_6
S_{10}^1	$V_{41}^1(5)$	t_1, t'_6	t_3, t'_4	p_2, p_3, p'_5, p'_6
S_{10}^2	$V_{41}^2(5)$	t_1, t'_4, t'_6	t_3, t'_3,t'_5	p_2, p_3, p'_4, p'_6
S_{10}^3	$V_{41}^3(5)$	t_1, t_3,t'_4,t'_6	t_2,t_4, t'_3,t'_5	p_2, p_4, p'_4, p'_6
S_{10}^4	$V_{41}^4(5)$	t_1, t_3, t'_6	t_2, t_4, t'_4	p_2, p_4, p'_5, p'_6

V_{ij} indicates V is the j-th i-dependent critical siphon. V^k ($k > 1$) indicates V is a EUP

Theorem 4.3

The controller obtained by Algorithm 4.1 is live and maximally permissive for β-nets.

Proof : Recall that two kinds of forbidden markings exist. Once they are controlled, all forbidden markings are avoided and the controlled net is live based on the theory in [8]. $H(V)$ has been shrinked so that $M_{max}(\Psi) = M_0(R(\Psi))$. One cannot insert a token into Ψ from a forbidden one to make it a live one (and yet still forbidden by V). Thus, no live states are forbidden and the maximal permissiveness is achieved. □

Theorem 4.4

The time complexity for Algorithm 4.1 is $O(|P_R|^2((|T| + |P_R|)|P_A^2|)|T|)$, where $|P_R|$ (resp. $|T|$, $|P_A|$) is the number of resource places (resp. transitions, activity places) in the net.

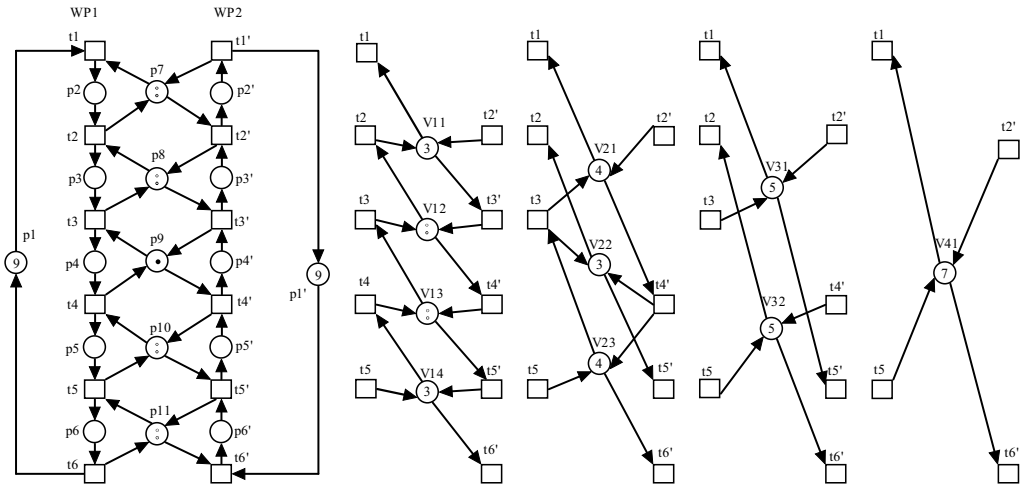
Proof : Step 1 takes $O(n)|T|$ time as shown in [10] since there are n basic siphons and there are $|T|$ control arcs. There are $O(k)$ boundary places for a k-dependent siphon. The number of EUP for a boundary place p_b is $O(|p_b^\bullet| + |P_R|)$. As analyzed earlier, for each boundary place, the number of possible pairs of (p, p') (to be considered for weighted control arcs) is $O(|P_A^2|)$. Thus, Step 2.1.1 takes $O(n^2|p_b^\bullet| + |P_R||P_A^2|)$ or $O(n^2|T| + |P_R||P_A^2|)$ time. There are $O(|T|)$ control arcs, hence ; Step 2.1.1 takes $O((n^2|T| + |P_R||P_N^2|)|T|)$ time. But $n{=}O(|P_R|)$; hence the time complexity for the algorithm is $O(|P_R|^2((|T| + |P_R|)|P_A^2|)|T|)$. □

Remarks : Note that Algorithm 4.1 takes polynomial amount of time much better than current maximally permissive control policies which are NP-hard and requires constructing reachability trees. Hence, our algorithm runs much faster. Also, the resulting controller is structually simple since no weighted control arcs are employed. The limitation is that for non-β nets, it may not be maximally permissive. It has been shown in [8] that any SMS can be synthesized from a strongly connected resource subnet (called core subnet). [8] also proposes to synthesize elementary (resp. dependent) siphons from resource, called basic circuits (resp. subnets). They are also called basic (resp. compound) siphons. It takes $O(|P_A| + |P_R|)$ time to compute all basic siphons in [8]. But it takes $O(|P_A|)$ time to find $[S]$ of all basic siphons to add monitors. Based on the theory in [69], $[S_0] = [S_1] \cup [S_2] \cup \ldots \cup [S_n]$. Hence, it takes $O(|P_A|)$ time to find $[S_0]$. Several basic siphons make up a compound siphon. It has been shown [8] that basic and compound siphons correspond to elementary and dependent siphons, respectively when the above basic circuits intersect at a single resource place. We propose in [96] to add monitors to each basic siphon built from elementary resource circuits [8] and find conditions for a compound siphon built from compound resource circuits to be already controlled. We show that if we assign monitors to basic siphons first, then many compound siphons are already controlled and need no monitors. The converse is not true. This avoids some redundant monitors and becomes more permissive. The presence of control places may induce new emptiable (called control) siphons. From these control siphons, one can derive mixture siphons. In [96], emptiable siphons are categorized into basic, compound, control and mixture siphons. If one carefully selects a sequence of emptiable siphons to add monitors, the number of monitors required can be reduced. This method does not need to enumerate all minimal siphons, nor to compute the reachability graph. Also no iterations are required and no need to remove redundant monitors. Hence, the computation burden is much less than

those by Uzam and Zhou (requires reachability analysis) as well as Piroddi *et al.* (requires MIP analysis). In addition, no control arcs are weighted. Furthermore, Lemma 4 in [10] indicates that it is relatively easy to identify basic circuits c_b between two neighboring working processes (WP). This plus Lemma 6 (all places in any c_b must be resource places) in [10] simplifies the search of c_b. There is no need to search circuits containing far-away resource places. Furthermore, it is easy to find c_b (normally formed among adjacent sharing resource places) when resource places between two adjacent WP are arranged in reverse order.

4.4 Examples

Example 1 : For the net in Fig. 4.2(a), there are four basic circuits from which, one can synthesize four basic siphons S_1, S_2, S_3, and S_4 and four 1- monitors V_{11}, V_{12}, V_{13}, and V_{14} are added accordingly as shown in Fig. 4.2(b). For each 2-dependent siphon, there is no boundary place and no exchange operation is available. Hence, the token pattern for an unmarked siphon follows that in Definition 4.3. Monitors V for these 2-dependent siphons are added (Fig. 4.2(c)) such that $[V] = [S]$. Figs 4.2(d) and (e) show the 3- and 4-monitors with $[V] = [S]$. The final controlled model reaches 2566 states.



(a) A Petri net (N', M'_0) (b) 1-monitor (c) 2-monitor (d) 3-monitor (e) 4-monitor

FIG. 4.2 – Another example of the control policy based on UP but with no unmarked places in $H(V)$.

To reach more states, consider the exchange operations for the 3-dependent siphons S'_1 and S'_2. $S_{R'_1} = \{p_8, p_9\}$ has only one boundary place p_9 with one exchange operation of moving a token from p'_3 to p'_4 such that the token pattern changes from $M'_\Psi = 2p_2 + 2(p_3 \oplus p'_3) + 2p'_5$ to $M'_{\Psi'} = 2p_2 + p_3 + p'_4 + 2p'_5$. A monitor V is added for each of Ψ and Ψ' with $M_0(V) = M_0(S) - \theta - 1 = 7 - 2 = 5$. $S_{R'_2} = \{p_9, p_{10}\}$ has only one boundary place p_9 with one exchange operation of moving a token from p_5 to p_4 so that the token pattern changes from $M'_{\Psi_*} = 2p_3 + 2(p_5 \oplus p'_5) + 2p'_6$ to $M'_{\Psi^\wedge} = 2p_3 + p_4 + p'_5 + 2p'_6$. Add a monitor V for each of Ψ^* and $\Psi^\wedge$ with $M_0(V) = M_0(S) - \theta - 1 = 7 - 2 = 5$. These two exchange operations lead to 2606>2566 live states.

There is only one 4-dependent siphon S. $S_R = \{p_8, p_9, p_{10}\}$ has only one boundary place p_9 but with 2 possible exchange operations. First one moves a token from p'_3 to p'_4 such that the token pattern changes from $M'_\Psi = 2p_2 + 2(p_3 \oplus p'_3) + 2(p_5 \oplus p'_5) + 2p'_6$ to $M'_{\Psi'} = 2p_2 + p_3 + p'_4 + 2(p_5 \oplus p'_5) + 2p'_6$. Add a monitor V for each of Ψ and Ψ' with $M_0(V) = M_0(S) - \theta - 1 = 9 - 2 = 7$. Note that when $M_0(p_9) = 2$, all internal places are nonsingular and all critical siphons are compound ones. The marking for each unmarked S is a UP.

Second one moves a token from p_5 to p_4 (this does not change the total number of tokens in $[S]$) such that the token pattern changes from $M'_\Psi = 2p_2 + 2(p_3 \oplus p'_3) + 2(p_5 \oplus p'_5) + 2p'_6$ to $M'_{\Psi^\wedge} = 2p_2 + 2(p_3 \oplus p'_3) + p_4 + p'_5 + 2p'_6$. Add a monitor V for Ψ with $M_0(V) = M_0(S) - \theta - 1 = 9 - 2 = 7$. There are 3 monitors added above. The resulting model as shown in Table 4.3 is maximally permissive and reaches 2628 states.

Note that for non-β nets of practical problems, most compound siphons fit the pattern in Definition 4.1. One can apply the method in this chapter to add monitors to the corresponding n-dependent siphons without computing them. The following two examples illustrate this point.

Example 2 : Now apply the approach to a well-known S^{3}PR [49] as shown in Fig. 4.3. We add a monitor for each basic siphon (Tables 4.4 and 4.5). As shown in Table 4.4, for the set of 2-dependent siphons related to S_4 synthesized from $c_4 = c_1 \circ c_2$, the unmarked pattern is $UP = p_3 \oplus p_{12} + p_2 \oplus p_{11} + p_4$. This corresponds to that of compound siphon S_4. A monitor $V = V_4$ is added with $[V]$=$[S_4]$ and $M_0(V) = M(p_{15}) + M(p_{18}) - 1$=3 (see Table 4.5). Note that S_4 is non-uniform and no control siphon can be formed from V_1 and V_2. Hence, the compound siphon is the critical one consistent with that obtained from the UP.

TAB. 4.3 – Controlled model for the net in Fig. 4.2.

S	$V(M_0)$	$V_S^\bullet$	${}^\bullet V_S$	$[V_S]$
S_1	$V_{11}(3)$	t_1, t'_3	t_2, t'_2	p_2, p'_3
S_2	$V_{12}(2)$	t_2, t'_4	t_3, t'_3	p_3, p'_4
S_3	$V_{13}(2)$	t_3, t'_5	t_4, t'_4	p_4, p'_5
S_4	$V_{14}(3)$	t_4, t'_6	t_5, t'_5	p_5, p'_6
S_5	$V_{21}(4)$	t_1, t'_4	t_3, t'_2	p_2, p_3, p'_3, p'_4
S_6	$V_{22}(3)$	t_2, t'_5	t_3, t'_4	p_3, p'_5
S_7	$V_{23}(4)$	t_3, t'_6	t_5, t'_4	p_4, p_5, p'_5, p'_6
S_8^1	$V_{31}^1(5)$	t_1, t'_5	t_3, t'_3	p_2, p_3, p'_4, p'_5
S_8^2	$V_{31}^2(5)$	t_1, t'_3, t'_5	t_3, t'_2, t'_4	p_2, p_3, p'_3, p'_5
S_9^1	$V_{32}^1(5)$	t_2, t'_6	t_4, t'_4	p_3, p_4, p'_5, p'_6
S_9^2	$V_{32}^2(5)$	t_2, t_4, t'_6	t_3, t_5, t'_4	p_3, p_5, p'_5, p'_6
S_{10}^1	$V_{41}^1(7)$	t_1, t_4, t'_6	t_3, t_5, t'_3	$p_2, p_3, p_5, p'_4, p'_5, p'_6$
S_{10}^2	$V_{41}^2(7)$	t_1, t_4, t'_3, t'_6	t_3, t_5, t'_2, t'_4	$p_2, p_3, p_5, p'_3, p'_5, p'_6$
S_{10}^3	$V_{41}^3(7)$	t_1, t'_3, t'_6	t_4, t'_2, t'_4	$p_2, p_3, p_4, p'_3, p'_5, p'_6$

V_{ij} indicates V is the j-th i-dependent critical siphon. V^k $(k > 1)$ indicates V is a EUP

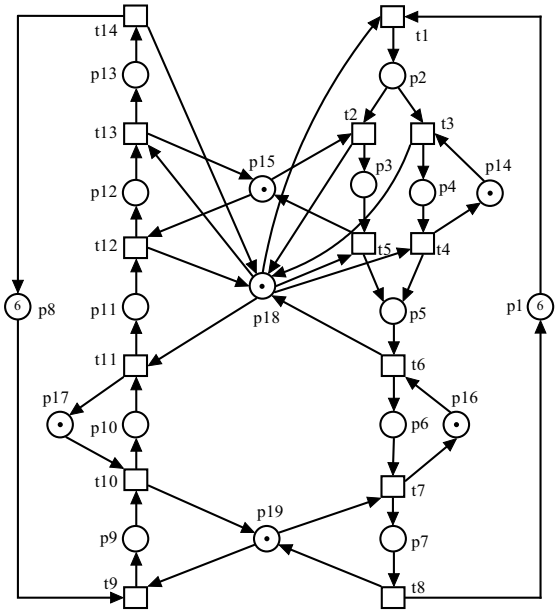


FIG. 4.3 – A more complicated Petri net model of an AMS.

For the set of 2-dependent siphons related to S_6 synthesized from $c_6 = c_2 \circ c_3$, the UP is UP=$p_3 + p_6 + p_9 + p_{10}$. This corresponds to that of partial mixture ($p - M$) siphon S_8. A monitor $V = V_8$ is added with $[V]$=|UP| and $M_0(V) = M(\text{UP}) - 1$=3. Note that $p_{12} \in (H(p_{15}) \cap [S_6])$, yet p_{12} is not in the above UP since no control place V exists such that $p_{12} \in^\bullet (V^\bullet)$. The corresponding EUP is $p_3 + p_5 + p_9 + p_{10}$, which necessarily reaches the above UP by firing t_6.

Finally, for the set of 3-dependent siphons related to the compound siphon synthesized from $c_1 \circ c_2 \circ c_3$, the UP is UP=$p_2 \oplus p_5 + p_3 + p_4 + p_6 + p_9 + p_{10}$. This corresponds to that of a partial mixture siphon S_{10} synthesized from the core subnet c_9+$[t_{11}p_{17}t_{10}p_{19}t_7V_{S_3}]$ obtained by adding TP-handle $[t_{11}p_{17}t_{10}p_{19}t_7V_{S_3}]$ upon the core circuit c_9. One can see that the condition ([96]) for the critical siphon to be a partial mixture siphon is : (a) The core circuit c contains two control places added for two siphons synthesized from two core circuits intersecting at a resource place r with $M_0(r) = 1$. (b) The corresponding control siphon is not emptiable. (c) The corresponding partial mixture siphon is emptiable.

Tab. 4.4 – Types of siphons, their dependency on basic siphons, $[S]$, UP, and EUP for the net in Fig. 4.3

Types	places	c	$[S]$ or UP or EUP	#
basic	$p_2, p_5, p_{11}, p_{13}, p_{14}, p_{18}$	$[p_{14}t_3p_{18}t_4]$		S_1
basic	$p_2, p_5, p_{13}, p_{15}, p_{18}$	$[p_{15}t_{12}p_{18}t_5]$	UP= $p_3 \oplus p_{12} + p_{11}$	S_2
basic	$p_2, p_7, p_{11}, p_{13}, p_{16}, p_{17}, p_{18}, p_{19}$	$[p_{16}t_6p_{18}t_{11}p_{17}t_{10}p_{19}t_7]$	UP= $p_5 + p_6 + p_9 + p_{10}$	S_3
comp.	$p_5, p_{13}, p_{14}, p_{15}, p_{18}$	$c_1 \circ c_2$	UP= $p_3 \oplus p_{12} + p_2 \oplus p_{11} + p_4$	S_4
comp.	$p_2, p_7, p_{11}, p_{13}, p_{14}, p_{16}, p_{17}, p_{18}, p_{19}$	$c_1 \circ c_3$	$[S] = \{p_3, p_4, p_5, p_6, p_9, p_{10}\}$	S_5
comp.	$p_2, p_4, p_7, p_{13}, p_{15}, p_{16}, p_{17}, p_{18}, p_{19}$	$c_2 \circ c_3$	$[S] = \{p_3, p_5, p_6, p_9, p_{10}, p_{11}, p_{12}\}$	S_6
contr.	$p_5, p_6, p_{11}, p_{12}, V_{S_2}, V_{S_3}$	$[V_{S_2}t_{11}V_{S_3}t_5]$	$[S] = \{p_3, p_9, p_{10}\} \in \pounds(S_6)$	S_7
$p - M$	$p_7, p_{11}, p_{12}, p_{17}, p_{19}, V_{S_2}, V_{S_3}$	c_7+$[t_{11}p_{17}t_{10}p_{19}t_7V_{S_3}]$	UP= $p_3 + p_6 + p_9 + p_{10} \in \pounds(S_6)$	S_8
$p - M$	$p_7, p_{11}, p_{12}, p_{17}, p_{19}, V_{S_2}, V_{S_3}$	c_7+$[t_{11}p_{17}t_{10}p_{19}t_7V_{S_3}]$	EUP= $p_3 + p_5 + p_9 + p_{10} \in \pounds(S_6)$	S'_8
contr.	$p_5, p_6, p_{11}, p_{12}, V_{S_3}, V_{S_4}$	$[V_{S_4}t_{11}V_{S_3}t_5]$	$[S] = \{p_2, p_3, p_4, p_9, p_{10}\} \in \pounds(S_4 \circ S_3)$	S_9
$p - M$	$p_7, p_{11}, p_{12}, p_{17}, p_{19}, V_{S_3}, V_{S_4}$	c_9+$[t_{11}p_{17}t_{10}p_{19}t_7V_{S_3}]$	UP= $p_2 \oplus p_5 + p_3 + p_4 + p_6 + p_9 + p_{10} \in \pounds(S_4 \circ S_3)$	S_{10}

$p - M$ stands for partial mixture siphon ;
comp. stands for compound siphon ;
contr. stands for control siphon.

TAB. 4.5 – Controlled model for the net in Fig. 4.3.

S	$V_S^\bullet$	${}^\bullet V_S$	$V(M_0)$	$[V_S]$
S_2	t_2, t_{11}	t_5, t_{13}	V_{S_2} (1)	p_3, p_{11}, p_{12}
S_3	t_4, t_5, t_9	t_7, t_{11}	V_{S_3} (3)	p_5, p_6, p_9, p_{10}
S_4	t_1, t_{11}	t_4, t_5, t_{13}	V_{S_4} (2)	$p_2, p_3, p_4, p_{11}, p_{12}$
S_8	t_2, t_6, t_9	t_5, t_7, t_{11}	$V^1_{S_8}$ (3)	p_3, p_6, p_9, p_{10}
S'_8	t_2, t_4, t_9	t_6, t_{11}	$V^2_{S_8}$ (3)	p_3, p_5, p_9, p_{10}
S_{10}	t_1, t_9	t_7, t_{11}	$V^1_{S_{10}}$(4)	$p_2, p_3, p_4, p_5, p_6, p_9, p_{10}$

Example 3 : Next apply the approach to a second well-know S^{3}PR [23] as shown in Fig. 4.4. We add a monitor for each basic siphon (Tables 4.6 and 4.8). As shown in Table 4.7, for the set of 2-dependent siphons related to S_{15} synthesized from $c_{15} = c_1 \circ c_{16}$, the UP is $M(p_{21}) = M(p_{22}) = M(p_{26})$=0 and $M(p_{12}) = M_0(p_{21}) = M(p_{19}) = M_0(p_{22})$ =1, $M(p_{13}) + M(p_{18}) = M_0(p_{26})$ =2. This corresponds to that of compound siphon S_{15}. A monitor $V = V_7$ is added with $[V]$=$[S_{15}]$ and $M_0(V) = M(p_{12}) + M(p_{19}) + M(p_{13}) + M(p_{18}) - 1$=3. Note that $S_1 \cap S_{16} = \{p_{26}\}$ and $M_0(p_{26}) > 1$ is the condition in [96] for the critical siphon to be a compound one. If $M_0(p_{26}) = 1$, then the critical siphon is a control siphon as shown below, where S_{14} is synthesized from $c_{14} = c_{16} \circ c_{18}$, $S_{16} \cap S_{18} = \{p_{21}\}$ and $M_0(p_{21}) = 1$.

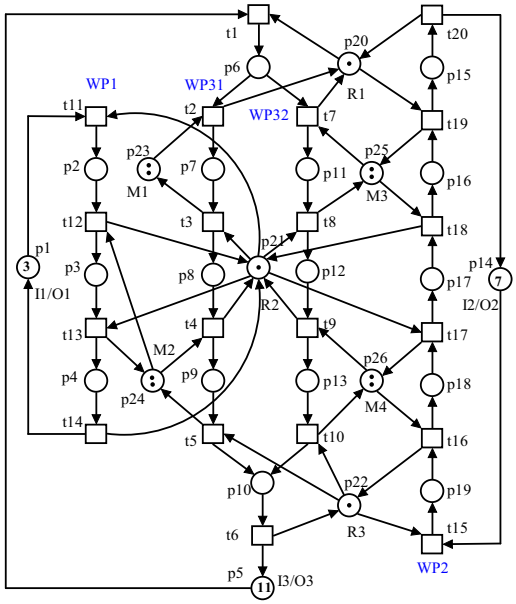


FIG. 4.4 – An S^{3}PR model in [23].

The UP is $M(p_{21}) = M_0(p_{21})$= 1, $M(p_{25}) = M(p_{26})$=0 and $M(p_{11}) = M_0(p_{25}) = M(p_{18}) =$

Tab. 4.6 – Basic siphons, resource circuits, and $V(M_0)$ for the net in Fig. 4.4

S_b	c_b	V(M_0)
S_1={$p_{10}, p_{18}, p_{22}, p_{26}$}	$[p_{22}t_{10}p_{26}t_{16}p_{22}]$	V_1 (2)
S_4={$p_4, p_{10}, p_{17}, p_{21}, p_{22}, p_{24}, p_{26}$}	$[p_{21}t_{17}p_{26}t_{16}p_{22}t_5p_{24}t_4p_{21}]$	V_2 (2)
S_{10}={$p_4, p_9, p_{12}, p_{17}, p_{21}, p_{24}$}	$[p_{21}t_{13}p_{24}t_4p_{21}]$	V_3 (2)
S_{16}={$p_2, p_4, p_8, p_{13}, p_{17}, p_{21}, p_{26}$}	$[p_{21}t_{17}p_{26}t_9p_{21}]$	V_4 (2)
S_{17}={$p_2, p_4, p_8, p_{12}, p_{15}, p_{20}, p_{21}, p_{23}, p_{25}$}	$[p_{21}t_3p_{23}t_2p_{20}t_{19}p_{25}t_{18}p_{21}]$	V_5 (5)
S_{18}={$p_2, p_4, p_8, p_{12}, p_{16}, p_{21}, p_{25}$}	$[p_{21}t_8p_{25}t_{18}p_{21}]$	V_6 (5)

Tab. 4.7 – Compound siphons, their dependency on basic siphons, UP, and EUP for the net in Fig. 4.4

S_0(U or NU)	dependency	UP	EUP
S_2(U)	$S_2 = S_4 \circ S_{17}$	$p_6+2p_7+2p_9+2(p_{11}\oplus p_{16})+2(p_{13}\oplus p_{18})+p_{19}$	$p_6+2p_7+p_8+p_9+2(p_{11}\oplus p_{16})+2(p_{13}\oplus p_{18})+p_{19}$, $p_6+2p_7+2p_9+2(p_{11}\oplus p_{16})+p_{12}+p_{18}+p_{19}$, $p_6+2p_7+2p_9+p_{11}+p_{17}+2(p_{13}\oplus p_{18})+p_{19}$
S_3(NU)	$S_3 = S_4 \circ S_{18}$		
S_5(NU)	$S_5 = S_{10}\circ S_{16}\circ S_{17}$		
S_6(NU)	$S_6 = S_{10}\circ S_{16}\circ S_{18}$		
S_7(NU)	$S_7 = S_{10} \circ S_{16}$		
S_8(NU)	$S_8 = S_{10} \circ S_{17}$		
S_9(NU)	$S_9 = S_{10} \circ S_{18}$		
S_{11}(NU)	$S_{11} = S_1\circ S_{16}\circ S_{17}$		
S_{12}(NU)	$S_{12} = S_{16} \circ S_{17}$		
S_{13}(U)	$S_{13} = S_1\circ S_{16}\circ S_{18}$	$2p_{11}+2(p_{13}\oplus p_{18})+p_{19}$	$2p_{11}+p_{12}+p_{18}+p_{19}$
S_{14}(U)	$S_{14} = S_{16} \circ S_{18}$	$2p_{11}+2p_{18}$	
S_{15}(U)	$S_{15} = S_1 \circ S_{16}$	$p_{12}+2(p_{13}\oplus p_{18})+p_{19}$	

Tab. 4.8 – Controlled model for the net in Fig. 4.4.

S	$V(M_0)$	$V_S^\bullet$	${}^\bullet V_S$	$[V_S]$
S_1	V_1 (2)	t_9, t_{15}	t_{10}, t_{16}	p_{13}, p_{19}
S_{10}	V_2 (2)	t_3, t_{11}	t_4, t_{13}	p_2, p_3, p_8
S_{18}	V_3 (2)	t_7, t_{17}	t_8, t_{18}	p_{11}, p_{17}
S_{16}	V_4 (2)	t_8, t_{16}	t_9, t_{17}	p_{12}, p_{18}
S_{15}	V_5 (3)	t_8, t_{15}	t_{10}, t_{17}	$p_{12}, p_{13}, p_{18}, p_{19}$
S_{19}	V_8 (3)	t_7, t_{16}	t_8, t_{17}	p_{11}, p_{18}
S_4	V_6 (5)	t_3, t_8, t_{11}, t_{15}	$t_5, t_{10}, t_{13}, t_{17}$	$p_2, p_3, p_8, p_9, p_{12}\ p_{13}, p_{18}, p_{19}$
S_{17}	V_7 (5)	t_1, t_{17}	t_3, t_8, t_{19}	$p_6, p_7, p_{11}, p_{16}, p_{17}$
S_{13}^1	V_9^1 (4)	t_7, t_{15}	t_9, t_{17}	$p_{11}, p_{12}, p_{18}, p_{19}$
S_{13}^2	V_9^2 (4)	t_7, t_9, t_{15}	t_8, t_{10}, t_{17}	$p_{11}, p_{13}, p_{18}, p_{19}$
S_{21}	V_{10} (6)	t_3, t_7, t_{11}, t_{15}	t_5,t_8, t_{13},t_{17}	$p_2, p_3, p_8, p_9, p_{11}, p_{17}, p_{18}, p_{19}$
S_{22}^1	V_{11}^1 (9)	t_1, t_9, t_{15}, t_{18}	$t_5, t_8, t_{10}, t_{17}, t_{19}$	$p_6, p_7, p_8, p_9, p_{11}\ p_{13}, p_{16}, p_{18}, p_{19}$
S_{22}^2	V_{11}^2 (9)	t_1, t_4, t_9, t_{15}	$t_3, t_5, t_8, t_{10}, t_{18}$	$p_6, p_7, p_9, p_{11}, p_{13}\ p_{17}, p_{18}, p_{19}$
S_{22}^3	V_{11}^3 (9)	t_1, t_4, t_{15}, t_{18}	$t_3, t_5, t_9, t_{17}, t_{19}$	$p_6, p_7, p_9, p_{11}, p_{12}, p_{16}, p_{18}, p_{19}$

$M_0(p_{26})$ =2. This corresponds to that of a control siphon $S_{19} = \{V_6, V_4, p_{12}, p_{17}\}$. A monitor $V = V_8$ is added with $[V]$=$[S_{19}]$ ={p_{11}, p_{18}} and $M_0(V) = M(p_{11}) + M(p_{18}) - 1$=3.

For the set of 3-dependent siphons related to S_{13} synthesized from $c_{13} = c_1 \circ c_{16} \circ c_{18}$, the UP is $M(p_{21}) = M_0(p_{21})$= 1, $M(p_{22}) = M(p_{25}) = M(p_{26})$=0 and $M(p_{11}) = M_0(p_{25}) = 2, M(p_{19}) = M_0(p_{22})$=1, $M(p_{13}) + M(p_{18}) = M_0(p_{26})$ =2. This corresponds to that of a partial mixture siphon $S_{20} = \{V_7, V_4, p_{10}, p_{17}\}$ synthesized from the core subnet obtained by adding TP-handle $[t_{17}V_8t_{16}p_{22}t_{10}V_7]$ upon the core circuit $[t_8V_6t_{17}V_7t_8]$.

A monitor $V = V_9^1$ is added with $[V]$=$[S_{20}]$ ={$p_{11}, p_{12}, p_{13}, p_{18}, p_{19}$} and $M_0(V) = M(p_{11}) + M(p_{13}) + M(p_{18}) + M(p_{19}) - 1$=4. There is only one EUP : $2p_{11} + p_{12} + p_{18} + p_{19}$, for which we add monitor V_9^2.

For the set of 2-dependent siphons related to S_3 synthesized from $c_3 = c_4 \circ c_{18}$, the pattern rule no longer holds since it is not uniform (i.e., NU in Column 2 of Table 4.7) : c_{18} (resp. c_4) spans between *WP32* (resp. *WP31*) and *WP2* (see Fig. 4.4).

The critical siphon S_{21} ={V_8, p_{21}, p_{22}, p_{24}, p_{17}, p_{10}} is synthesized from the core circuit $c = [V_8t_{16}p_{22}t_5p_{24}t_4p_{21}t_8V_8]$ that contains transitions on *WP31*, *WP32*, and *WP2*. Note p_{21} is no longer an internal resource place here since it is on the above resource circuit c. The UP is $M(p_{21}) = M(p_{22}) = M(p_{24}) = M(p_{25})$=0, $M(p_{26})$ =1, and $M(p_9) = M_0(p_{24}) = M(p_{11})$ =

$M_0(p_{25}) = 2, M(p_{18}) = M(p_{19}) = M_0(p_{22}) = 1$, and $M(p_2) + M(p_3) + M(p_8) = M_0(p_{21})$=1. A monitor $V = V_{10}$ is added with $[V]$=$[S_{21}]$ ={p_2, p_3, p_8, p_9, p_{11}, p_{18}, p_{19}}and $M_0(V) = M(p_2) + M(p_3) + M(p_8) + M(p_9) + M(p_{11}) + M(p_{18}) + M(p_{19}) - 1 = 6 = M_0(R(S)) - \theta - 1$, θ=1 consistent with the UP since the presence of V_8 reduces the maximal marking in $\{p_{11}, p_{18}\}$ by one.

For the set of 2-dependent siphons related to S_2 synthesized from $c_2 = c_4 \circ c_{17}$, the UP is $M(p_{21}) = M_0(p_{21})$= 1, $M(p_{22}) = M(p_{23}) = M(p_{24}) = M(p_{25}) = M(p_{26})$=0 and $M(p_6) = M_0(p_{20}) = M(p_{19}) = M_0(p_{22})$ =1, $M(p_{13}) + M(p_{18}) = M_0(p_{26})$ =2, $M(p_{11}) + M(p_{16}) = M_0(p_{25})$ =2, $M(p_7) = M_0(p_{23}) = M(p_9) = M_0(p_{24})$ =2. This corresponds to that of a full mixture siphon S_{22} ={V_5, V_2, p_{10}, p_{15}} synthesized from the core subnet obtained by adding 2 TP-handles $[t_3 p_{23} t_2 p_{20} t_{19} V_5]$, $[t_{17} p_{26} t_{16} p_{22} t_5 V_2]$, and $[t_{17} p_{26} t_{16} p_{22} t_{10} V_2]$ upon the core circuit $[V_2 t_8 V_5 t_{17} V_2]$. It is called a full mixture siphon since one cannot synthesize any new SMS by adding more handles. A monitor $V = V_{11}^1$ is added with $[V]$=$[S_{22}]$ ={$p_6, p_7, p_9, p_{11}, p_{13}, p_{16}, p_{18}, p_{19}$} and $M_0(V) = M(p_6) + M(p_7) + M(p_9) + M(p_{11}) + M(p_{13}) + M(p_{16}) + M(p_{18}) + M(p_{19}) - 1$=9.

There are three EUP : (1) $p_6 + 2p_7 + p_8 + p_9 + 2(p_{11} \oplus p_{16}) + 2(p_{13} \oplus p_{18}) + p_{19}$, (2) $p_6 + 2p_7 + 2p_9 + 2(p_{11} \oplus p_{16}) + p_{12} + p_{18}) + p_{19}$, and (3) $p_6 + 2p_7 + 2p_9 + p_{11} + p_{17} + 2(p_{13} \oplus p_{18}) + p_{19}$.

Note that the set of unmarked activity places for UP is a proper subset of that for EUP (1). By Theorem 2 in [12], the monitor $V = V_{11}^1$ for the UP is redundant. A monitor (V_{11}^1, V_{11}^2, and V_{11}^3) is added for the above EUP. And there are live states that cannot be reached caused by V_{11}^1.

The resulting model in Table 4.8 is live where we have added 14 monitors and 82 control arcs vs 19 monitors and 120 control arcs in [11] but with 21,562 good states, which is slightly less than the maximally permissive one (21,581) in [73], but with much faster computation and no weighted control arcs.

4.5 Summary

This chapter proposes a maximally permissive control policy for a subclass of S^3PR based on the new ground-breaking theory of token distribution pattern of unmarked siphons. This has the advantage of avoiding the computation of new siphons derived from monitor places since the UP solely determines the controller region (or control arcs) and the initial marking. This is no need to construct the reachability tree and hence the problem is no longer NP-hard. This results in

fewer monitors and more reachable states as shown by two well-known S^3PRs. Future work should consider how to add minimal number of weighted control arcs to make controlled nets maximally permissive and to extend to other complicated nets.

The major contribution in this research is published :

[1] **Gaiyun Liu**, Daniel Yuh Chao, and Fang Yu, Control policy for a subclass of Petri nets without reachability analysis, *IET Control Theory and Applications*, vol.7, no.8, pp.1131-1141, 2013.

Chapitre 5

Robustness of Deadlock Control for S^3PR with Unreliable Resources

5.1 Introduction

A variety of deadlock control policies based on Petri nets have been proposed for AMS. Most of them prevent deadlocks by adding monitors for emptiable siphons that, without an appropriate control policy, can cause deadlocks, where the resources in a system under consideration are assumed to be reliable. When resources are unreliable, it is infeasible or impossible to apply the existing control strategies. For S^3PR, this chapter bridges the gap between a divide-and-conquer deadlock control strategy and its application to real-world systems with unreliable resources. Recovery subnets and monitors are designed for unreliable resources and strict minimal siphons that may be emptied, respectively. Normal and inhibitor arcs are used to connect monitors with recovery subnets in case of necessity. Then reanalysis of the original Petri net is avoided and a robust liveness-enforcing supervisor is derived. The supervisors designed for S^3PR by the proposed method has following properties : (1) they can prevent deadlocks for plant models when all resources work normally ; (2) deadlocks are prevented even if some resources fail to work and are removed to repair at any time ; and (3) waiting-for-repair states (see Definition 13) disappear after the repaired resources are returned.

The rest of this chapter is organized as follows. Section 2 motivates this study via an example. A design method of a robust liveness-enforcing supervisor for an S^3PR net with unreliable resources is proposed in Section 3. Examples are given in Section 4 to demonstrate the proposed

method. Section 5 discusses some open problems. Finally, Section 6 concludes this work and suggests directions for future research.

5.2 Motivation

An automated manufacturing cell shown in Fig. 5.1 (a) has a machine tool M and a type of robots R with two robots. Each of the robots can move and the machine tool can process one part at a time. Parts enter the cell through uploading buffer I, and leave the cell through downloading buffer O. The machine tool performs an activity on raw parts and robots deal with the movements of parts. The cell can be modeled with Petri nets as Fig. 5.1 (b) shows. The net is an S^3PR, where $P^0 = \{p_1\}$, $P_R = \{p_5, p_6\}$, and the others are activity places.

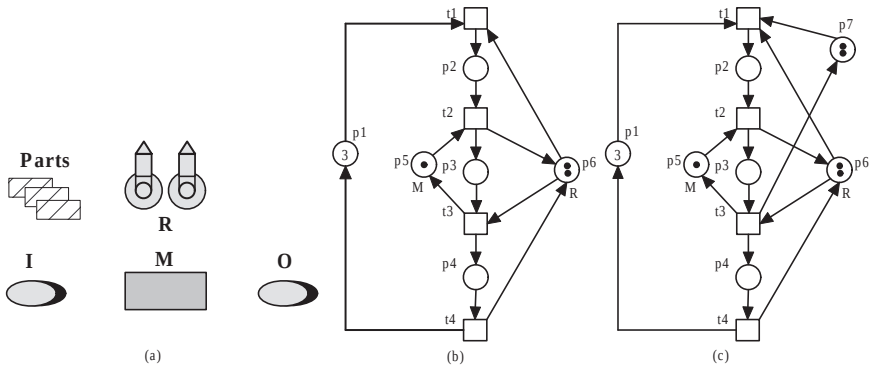


FIG. 5.1 – (a) An automated manufacturing cell, (b) a Petri net model, (c) a liveness-enforcing supervisor.

In Fig. 5.1 (b), $S = \{p_4, p_5, p_6\}$ is a unique SMS. The siphon is empty at marking $2p_2 + p_3$ and correspondingly the system is in a deadlock state. Physically, it means that each robot is holding and the machine tool is processing a part at the same time. All the resources are occupied and they are in a circular wait. We need to avoid this case such that the system can keep running. Using the traditional methods in [23], [47], and [69] that add monitors for siphons, we can find a liveness-enforcing supervisor for the S^3PR, as shown in Fig. 5.1 (c).

The deadlock control policies in [23], [47], and [69] are developed by assuming that the resources in a system are reliable. However, real-world AMS often suffer from unreliable resource failures. In Fig. 5.1 (a), we assume that one robot breaks down at the initial marking. Then the ro-

bot with a malfunction needs to be removed and recovered. One token is correspondingly removed from p_6 in Fig. 5.1 (c) and then a net shown in Fig. 5.2 is derived.

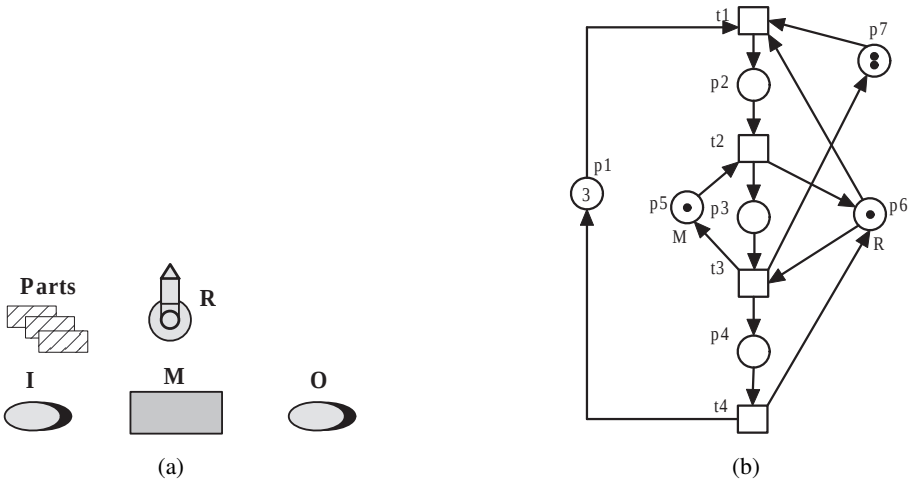


FIG. 5.2 – (a) One robot is removed, (b) one token is correspondingly removed from p_6.

In Fig. 5.2(b), marking $p_1 + p_2 + p_3$ is a deadlock state. Monitor p_7 cannot control the system any more in this case. In other words, the supervisor of the system is not robust in the sense that a broken robot leads the original controlled net system to a deadlock state. It is necessary to design a robust liveness-enforcing supervisor for the Petri net, which can control the system whether a robot breaks down or not.

As shown in Fig. 5.3, we remove a token from monitor p_7 when a token is removed from p_6, i.e., t_7 is fired. This means that the robot with a malfunction is sent to be repaired. In this case, monitor p_7 with one token makes the system controllable. In fact, for resource type R, robots may break down in p_2, p_4, or p_6. Once a resource breaks down in some place, it needs to be removed, which does not depend on whether its related monitor is marked or not. Based on this fact, a robust liveness-enforcing supervisor is designed as Fig. 5.3 shows. More detailed explanations will be shown in the next section.

In this chapter, we try to design a supervisor for a plant model such that deadlocks cannot occur even if some resources in a system break down.

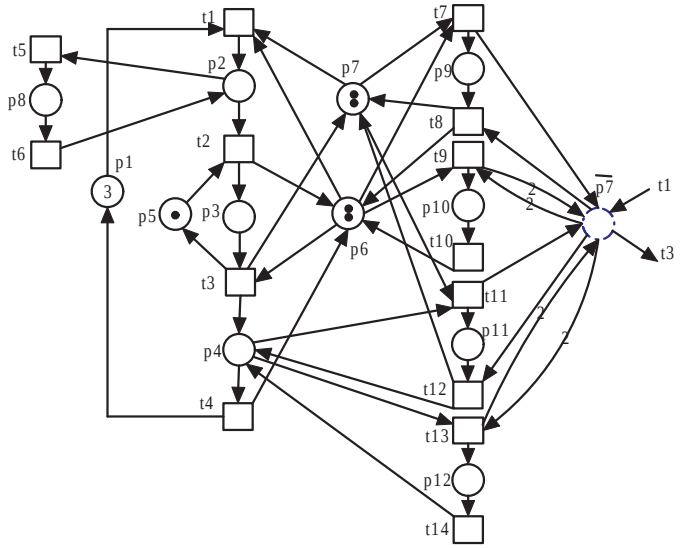


FIG. 5.3 – A robust supervisor for a system.

5.3 Robust Liveness-enforcing Supervisor Design

This section presents algorithms to design a robust liveness-enforcing supervisor for an S^3PR with unreliable resources.

Let (N, M_0) be an S^3PR plant model and (N_{sup}, M_{sup}) be its liveness-enforcing supervisor. The controlled system of the plant is denoted as (N_V^c, M_{V0}^c). The robustness of (N_{sup}, M_{sup}) is a system property to keep the controlled system live as some resources break down. In this chapter, we aim to design supervisors for S^3PR with the following properties : (1) they can prevent deadlocks for plant models when all resources work normally ; (2) deadlocks are prevented even if some resources fail to work and are removed to repair at any time ; and (3) waiting-for-repair states (see Definition 13) disappear after the repaired resources are returned. Hence, the supervisors are more robust than the traditional ones in [23], [47], and [69]. They can self-adaptively control the deadlocks without reanalysis.

Specifically, this chapter develops a two-stage approach to synthesize robust liveness-enforcing supervisors for AMS that can be modeled with S^3PR. First, we find a liveness-enforcing supervisor for an S^3PR by the divide-and-conquer strategy proposed in [76]. Second, to improve the robustness of the supervisor, we add recovery subnets, complementary places of monitors, and necessary

arcs to the system obtained by the first stage such that liveness is still preserved. We find that the complementary places of monitors and related arcs can be replaced by inhibitor arcs. Then, an improved algorithm is reported.

It is worthy of noting that the purposes of the divide-and-conquer strategy proposed in [76] and the strategy in this chapter are different. The former is a deadlock control strategy that guarantees the liveness of the original system. The latter focuses on a robust control strategy that improves the robustness of the supervisor of the original system with unreliable resources.

5.3.1 Liveness-enforcing Supervisor Design

5.3.1.1 A Classical Deadlock Prevention Policy

In this section, we mainly review the deadlock prevention policy proposed by Ezpeleta *et al*. in [23], which is used in the divide-and-conquer strategy in the next section. It develops a systematic method to establish a liveness-enforcing supervisor for an S^3PR by adding monitors for its SMS such that they are prevented from being emptied. Some useful notations [75] are first introduced as follows :

- Let C be a circuit of N and x and y be two nodes of C. Node x is said to be previous to y if there exists a path in C from x to y, the length of which is greater than one and does not pass over the idle process place p^0. This fact is denoted as $x <_C y$.
- Let x and y be two nodes in N. Node x is said to be previous to y in N if there exists a circuit C such that $x <_C y$. This fact is denoted by $x <_N y$.
- Let x and $A \subseteq P \cup T$ be a node and a set of nodes in N, respectively. Then $x <_N A$ iff there exists a node $y \in A$ such that $x <_N y$ and $A <_N x$ iff there exists a node $y \in A$ such that $y <_N x$.

Take the net shown in Fig. 5.4 (a) as an example. In this net, $C{=}p_1t_1p_7t_2p_3t_3p_5t_4p_6t_5p_1$ is a circuit and $EP(p_7, p_6) = p_7t_2p_3t_3p_5t_4p_6$ is a path in C. The support of $EP(p_7, p_6)$ is $\{p_7, t_2, p_3, t_3, p_5, t_4, p_6\}$ and the support of C is $\{p_1, t_1, p_7, t_2, p_3, t_3, p_5, t_4, p_6, t_5\}$. Clearly, we have $p_7 <_C p_6$ and $p_7 <_N p_6$.

The following notations are also useful in the establishment of the deadlock prevention policy in [23]. Mathematically, given a set A, the power set of A, written as 2^A, is the set of all subsets of

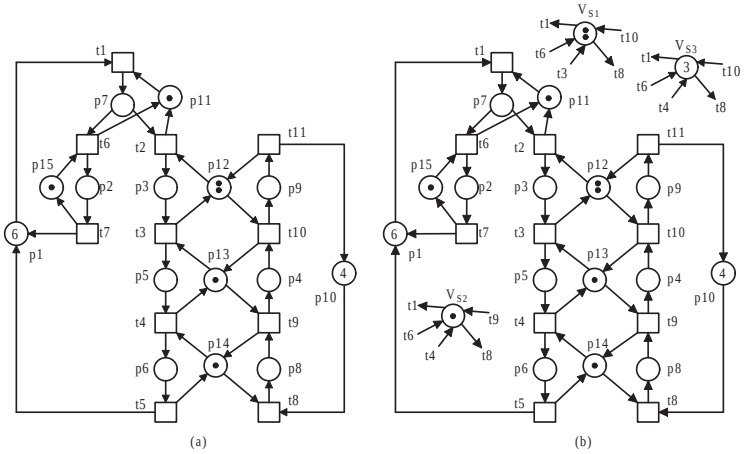


FIG. 5.4 – An S^3PR (N, M_0).

A. Note that Π is used to denote the set of SMS in an S^3PR (N, M_0). The sets of downstream and upstream siphons of a transition are defined as follows.

Definition 5.1 *[75] Let $\Delta^+(t)$ ($\Delta^-(t)$) denote the set of downstream (upstream) siphons of a transition t and $\mathcal{P}_S$ denote the adjoint set of a siphon S in an S^3PR $N = \bigcirc_{i=1}^{n} N_i = (P^0 \cup P_A \cup P_R, T, F)$.*

(1) $\Delta^+ : T \to 2^\Pi$ is a mapping : If $t \in T_i$, then $\Delta^+(t) = \{S \in \Pi | t <_{\overline{N}_i} [S]^i\}$. If $S \in \Delta^+(t)$, then the set $[S]^i$ is reachable *from t, i.e., there exists a path in $\overline{N}_i$ leading from t to an activity place $p \in P_{A_i}$ that is not included in S but uses a resource of S, where $[S] = \cup_{i=1}^{n} [S]^i$, $P_A = \cup_{i=1}^{n} P_{A_i}$, and $[S]^i = [S] \cap P_{A_i}$.*

(2) $\Delta^- : T \to 2^\Pi$ is a mapping : If $t \in T_i$, then $\Delta^-(t) = \{S \in \Pi | [S]^i <_{\overline{N}_i} t\}$.

(3) $\forall i \in \mathbb{N}_n$, $\forall S \in \Pi$, $\mathcal{P}_S^i = [S]^i \cup \{p \in P_{A_i} | p <_{\overline{N}_i} [S]^i\}$, and $\mathcal{P}_S = \cup_{i=1}^{n} \mathcal{P}_S^i$.

Take the net shown in Fig. 5.4 (a) as an example. There are three SMS $S_1 = \{p_5, p_9, p_{12}, p_{13}\}$, $S_2 = \{p_4, p_6, p_{13}, p_{14}\}$, and $S_3 = \{p_6, p_9, p_{12}, p_{13}, p_{14}\}$. Their complementary sets are $[S_1] = \{p_3, p_4\}$, $[S_2] = \{p_5, p_8\}$, and $[S_3] = \{p_3, p_4, p_5, p_8\}$, respectively. We have downstream siphons $\Delta^+(t_1) = \Delta^+(t_2) = \Delta^+(t_8) = \{S_1, S_2, S_3\}$, $\Delta^+(t_3) = \{S_2, S_3\}$, $\Delta^+(t_9) = \{S_1, S_3\}$ and $\Delta^+(t_4) = \Delta^+(t_{10}) = \emptyset$. Similarly, upstream siphons include $\Delta^-(t_1) = \Delta^-(t_2) = \Delta^-(t_6) = \Delta^-(t_7) = \emptyset$, $\Delta^-(t_3) = \{S_1\}$, and $\Delta^-(t_4) = \Delta^-(t_5) = \{S_1, S_2, S_3\}$.

We have adjoint sets $\mathcal{P}_{S_1} = \mathcal{P}^1_{S_1} \cup \mathcal{P}^2_{S_1} = (\{p_3\} \cup \{p_7\}) \cup (\{p_4\} \cup \{p_8\}) = \{p_3, p_4, p_7, p_8\}$, $\mathcal{P}_{S_2} = \mathcal{P}^1_{S_2} \cup \mathcal{P}^2_{S_2} = (\{p_5\} \cup \{p_7, p_3\}) \cup \{p_8\} = \{p_7, p_3, p_5, p_8\}$, and $\mathcal{P}_{S_3} = \mathcal{P}^1_{S_3} \cup \mathcal{P}^2_{S_3} = (\{p_3, p_5\} \cup p_7) \cup \{p_4, p_8\} = \{p_7, p_3, p_5, p_4, p_8\}$.

Definition 5.2 *[75] Let (N, M_0) be an S^3PR with $N = \bigcirc_{i=1}^{n} N_i = (P_A \cup P^0 \cup P_R, T, F)$. The net $(N_V, M_{0V}) = (P_A \cup P^0 \cup P_R \cup P_V, T, F \cup F_V, M_{0V})$ is the controlled system of (N, M_0) if*

(1) $P_V = \{V_S | S \in \Pi\}$ is a set of monitors such that there exists a bijective mapping between Π and P_V.

(2) $F_V = F^1_V \cup F^2_V \cup F^3_V$, where

$F^1_V = \{(V_S, t) | S \in \Delta^+(t), t \in P^{0\bullet}\}$,

$F^2_V = \{(t, V_S) | t \in [S]^\bullet, S \notin \Delta^+(t)\}$, *and*

$F^3_V = \cup_{i=1}^{n} \{(t, V_S) | t \in T_i \setminus P^{0\bullet}, S \notin \Delta^-(t), {}^\bullet t \cap P_{A_i} \subseteq \mathcal{P}^i_S, t \not< [S]^i\}$.

(3) M_{0V} is defined as follows : (3.1) $\forall p \in P_A \cup P^0 \cup P_R$, $M_{0V}(p) = M_0(p)$ and (3.2) $\forall V_S \in P_V$, $M_{0V}(V_S) = M_0(S) - 1$.

For a strict minimal siphon S, $M_{0V}(V_S)$ defined in Definition 5.2 ensures that the maximal number of tokens held by $\mathcal{P}_S$ is not more than $M_0(S)$. By Definition 5.1, $[S] \subseteq \mathcal{P}_S$ holds. Hence, S cannot be unmarked if a monitor V_S is added for it.

Theorem 5.1

[23] (N_V, M_{0V}) is live .

For the net shown in Fig. 5.4 (a), three monitors are needed to prevent three SMS from being emptied. We first take $S_1 = \{p_5, p_9, p_{12}, p_{13}\}$ as an example. Since $P^0 = \{p_1, p_{10}\}$, we have $P^{0\bullet} = \{t_1, t_8\}$. As a result, $\{(V_{S_1}, t_1), (V_{S_1}, t_8)\} \subseteq F^1_V$.

Due to $[S_1] = \{p_3, p_4\}$, $[S_1]^\bullet = \{t_3, t_{10}\}$. Note that $S_1 \notin \Delta^+(t_3)$ and $S_1 \notin \Delta^+(t_{10})$. We have $\{(t_3, V_{S_1}), (t_{10}, V_{S_1})\} \subseteq F^2_V$.

Next let us find the arcs related to V_{S_1} in F^3_V. We can obtain $(T_1 \setminus P^{0\bullet}) \cup (T_2 \setminus P^{0\bullet}) = \{t_2\text{-}t_7, t_9\text{-}t_{11}\}$, $\{t | S_1 \notin \Delta^-(t), t \in T\} = \{t_1, t_2, t_6\text{-}t_9\}$, $\{t | {}^\bullet t \cap P_{A_1} \subseteq \mathcal{P}^1_S\} \cup \{t | {}^\bullet t \cap P_{A_2} \subseteq \mathcal{P}^2_S\} = \{t_2, t_3, t_6, t_9, t_{10}\}$, and $\{t | t \not< [S_1]^1\} \cup \{t | t \not< [S_1]^2\} = \{t_3\text{-}t_7, t_{10}, t_{11}\}$. Hence, $(t_6, V_{S_1}) \in F^3_V$.

For siphons S_2 and S_3, monitors V_{S_2} and V_{S_3} can be added with $\{(V_{S_2}, t_1), (V_{S_2}, t_8), (V_{S_3}, t_1), (V_{S_3}, t_8)\} \subseteq F_V^1$, $\{(t_4, V_{S_2}), (t_9, V_{S_2}), (t_4, V_{S_3}), (t_{10}, V_{S_3})\} \subseteq F_V^2$, and $\{(t_6, V_{S_2}), (t_6, V_{S_3})\} \subseteq F_V^3$. The controlled system for (N, M_0) is shown in Fig. 5.4 (b) with 64 reachable states.

The deadlock prevention policy in [23] is usually considered to be one of the most significant contributions in the deadlock control area using a Petri net formalism [37]. However, on the one hand, all output arcs of the new monitors are added to the source transitions of the original model. This leads to the restriction of its behavior. On the other hand, if we design robust liveness-enforcing supervisors for some S^3PR based on this deadlock control policy, the final supervisors will be structurally complex (This disadvantage will be discussed in Section 6). In this study, we use the divide-and-conquer deadlock control strategy to improve this policy.

5.3.1.2 Divide-and-Conquer Deadlock Control Strategy

In this section, the main idea of the divide-and-conquer deadlock control strategy is briefly reviewed to understand this chapter. More details can be found in [76].

Let $N = (P_A \cup P^0 \cup P_R, T, F)$ be an S^3PR. According to the concept of resource circuits, we classify the resources in P_R into two classes (Algorithm 2 in [76]) : the ones each of which is associated with a resource circuit from which an SMS can be derived, and the ones that are not associated with SMS. The former is in $P_R^1 \cup P_R^2 \cup \cdots \cup P_R^k$ and the latter is in P_R^F, where $\forall i, j \in \mathbb{N}_k$, $i \neq j$, $P_R^i \cap P_R^j = \emptyset$ and $k \in \mathbb{N}_k$. Thus, $P_R^1 \cup P_R^2 \cup \cdots \cup P_R^k \cup P_R^F = P_R$. The allocation mechanism of the resources in P_R^F in a plant S^3PR net model cannot lead to deadlocks.

Take Fig. 5.4 (a) as an example. Resource p_{11} does not belong to a resource circuit. Resource p_{12} is associated with resource circuit $C(p_{12}) = p_{12}t_{10}p_{13}t_3$ from which an SMS can be derived. We hence have $P_R^1 = \{p_{12}, p_{13}\}$. Resource p_{13} in $C(p_{12})$ is associated with a new resource circuit $C(p_{13}) = p_{13}t_9p_{14}t_4$. No new resource circuit containing p_{14} can be found and no resource circuit is associated with resource p_{15}. Finally, we have $P_R^1 = \{p_{12}, p_{13}, p_{14}\}$ and $P_R^F = \{p_{11}, p_{15}\}$. Clearly, the use of resources p_{11} and p_{15} can not cause deadlocks.

Definition 5.3 *[76] Let $N = (P_A \cup P^0 \cup P_R, T, F)$ be an S^3PR. It is said to be disassemblable if $P_R^F \neq \emptyset$.*

This definition means that if the set of resources that are not associated with SMS is empty,

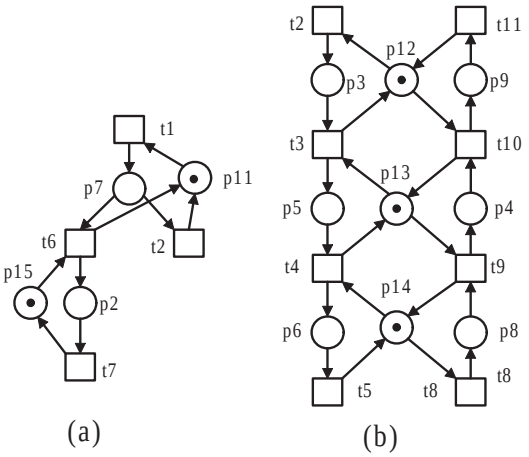


FIG. 5.5 – (a) An autonomous subnet, (b) a toparchy of the net in Fig. 5.4 (a).

the S^{3}PR can not be disassembled. In this case, the divide-and-conquer deadlock control strategy cannot take advantage of it. For example, the net shown in Fig. **??** can not be disassembled since $P_R^F = \emptyset$. While the net shown in Fig. 5.4 (a) is disassemblable since $P_R^F = \{p_{11}, p_{15}\} \neq \emptyset$.

Definition 5.4 *[76] A subnet (N^i, M_0^i) derived from $P^i \cup T^i$ in an S^3PR (N, M_0) with $N = (P^0 \cup P_A \cup P_R, T, F)$ is called a toparchy derived from P_R^i, where $P^i = P_R^i \cup \{p \in P_A | p \in H(r), r \in P_R^i\}$ and $T^i = \cup_{p \in P^i}({}^\bullet p \cup p^\bullet)$.*

Definition 5.5 *[76] The subnet (N^F, M_0^F) derived from $P^F \cup T^F$ in an S^3PR (N, M_0) with $N = (P^0 \cup P_A \cup P_R, T, F)$ 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 $T^F = \cup_{p \in P^F}({}^\bullet p \cup p^\bullet)$.*

For the net in Fig. 5.4 (a), we have $P_R^1 = \{p_{12}, p_{13}, p_{14}\}$ and $P_R^F = \{p_{11}, p_{15}\}$. It is easy to see that $P^1 = \{p_{12}, p_{13}, p_{14}\} \cup \{p_3, p_4, p_5, p_6, p_8, p_9\}$, $T^1 = \{t_2, t_3, t_4, t_5, t_8, t_9, t_{10}, t_{11}\}$, $P^F = \{p_{11}, p_{15}\} \cup \{p_2, p_7\}$, and $T^F = \{t_1, t_2, t_6, t_7\}$. As a result, the net in Fig. 5.4 (a) can be decomposed into a toparchy and an autonomous subnet, as shown in Fig. 5.5.

Theorem 5.2

[76] An autonomous subnet (N^F, M_0^F) is live.

Theorem 5.3

[76] A toparchy (N^i, M_0^i) is not live.

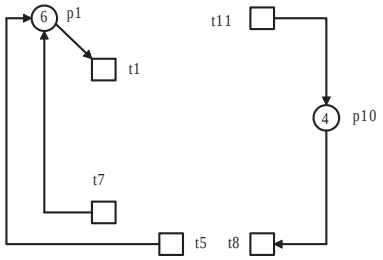


FIG. 5.6 – The idle subnet (N^{id}, M_0^{id}).

By Definition 5.5, an S^3PR has at most one unique autonomous subnet. From Theorem 5.2, an autonomous subnet (N^F, M_0^F) derived from P_R^F does not need to be controlled for its liveness. Indeed, to achieve the deadlock control purposes for an S^3PR, we need to consider its toparchies only.

Definition 5.6 *[76] Let (N, M_0) be an S^3PR with $N = (P^0 \cup P_A \cup P_R, T, F)$. A subnet derived from $(P^0, T_{SO} \cup T_{SI})$, denoted by (N^{id}, M_0^{id}), is called the idle subnet of (N, M_0), where $\forall p \in P^0$, $M_0^{id}(p) = M_0(p)$, T_{SO} and T_{SI} denote the set of source transitions and the set of sink transitions, respectively.*

Definition 5.7 *[76] A toparchy is said to be subordinate (dominate) is its idle-augmented net is live (not live).*

For example, the net shown in Fig. 5.6 is the idle subnet of (N, M_0) in Fig. 5.4 (a). Definition 5.7 is used in Section 6 to show why toparchies are not distinguished into subordinate or dominate ones in Algorithm 5.1.

Definition 5.8 *[76] Let (N_1, M_1) and (N_2, M_2) be two nets with $N_i = (P_i, T_i, F_i, W_i)$, $i = 1, 2$, satisfying $P_1 \cap P_2 = \emptyset$. (N, M) 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), denoted by $(N_1, M_1) \bigotimes (N_2, M_2)$, if*

(1) $P = P_1 \cup P_2$;

(2) $T = T_1 \cup T_2$;

(3) $F = F_1 \cup F_2$;

(4) $W = W_1 \cup W_2$;

(5) $M(p) = M_i(p)$, $p \in P_i$, $i = 1, 2$.

Definition 5.9 *[76] Let* (N_1, M_1), (N_2, M_2), $\cdots$, *and* (N_k, M_k) *be k nets satisfying* $P_i \cap P_j = \emptyset$, $\forall i, j \in \mathbb{N}_k$, $i \neq j$. *The synchronous synthesis of* (N_1, M_1), (N_2, M_2), $\cdots$, *and* (N_k, M_k) *is defined as* $(N, M) = (N_k, M_k) \bigotimes (\bigotimes_{i=1}^{k-1} (N_i, M_i))$.

Based on above definitions and theorems, a divide-and-conquer strategy to tackle the deadlock problems in a resource allocation system that is modeled with Petri nets is proposed in [76]. A plant net model is first disassembled into an idle subnet, an autonomous subnet if it exists and a number of toparchies. Then, a liveness-enforcing sub-supervisor, called toparch, is designed for each toparchy. Finally, the idle subnet, the autonomous subnet, and all toparches are merged into a net, called monarch that is shown to be a liveness-enforcing supervisor for the whole plant Petri net model. Specifically, the monarch of an S^3PR plant model (N, M_0) can be constructed using the following algorithm.

Algorithm 5.1

liveness-enforcing supervisor design [76]

Input : an S^3*PR plant model* (N, M_0)

Output : (N_V^c, M_{V0}^c)

begin{

compute the idle subnet (N^{id}, M_0^{id})

compute the autonomous subnet (N^F, M_0^F)

compute all toparchies (N^i, M_0^i), $i \in \mathbb{N}_k$

for $(i = 1 \,;\, i \leq k \,;\, i++)$ ***do***

design a $G_i = (N^{i\alpha}, M_0^{i\alpha})$ *for each toparchy* (N^i, M_0^i) *by the deadlock prevention policy designed by Definition 5.2*

end for

synchronously synthesize (N^{id}, M_0^{id}), (N^F, M_0^F), $G_1, G_2, \ldots$, *and* G_k

output the synthesized net $G = (N_V^c, M_{V0}^c)$

}end of the algorithm

Theorem 5.4

[76] The monarch $G = (N_V^c, M_{V0}^c)$ *synthesized by Algorithm 5.1 is a controlled Petri net for*

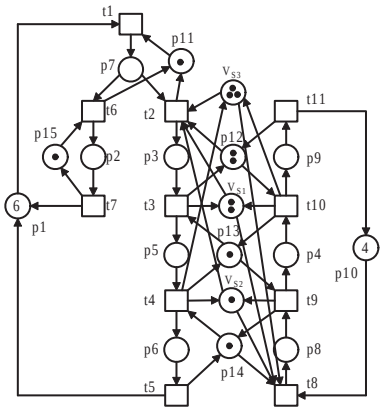


FIG. 5.7 – Monarch synthesis for (N, M_0) in Fig. 5.4 (a).

(N, M_0).

By Algorithm 5.1, the monarch synthesis for the plant net model in Fig. 5.4 (a) is obtained as Fig. 5.7 shows. This supervisor leads to 96 reachable states. Comparing Fig. 5.7 with Fig. 5.4 (b), we can find that the supervisor is with more reachable states but a less complex structure. However, when $P_R^F = \emptyset$, an S^3PR cannot be disassembled. In this case, Algorithm 5.1 is equivalent with the deadlock control policy described by Definition 5.2. For example, if Algorithm 5.1 is used to control the net model in Fig. 5.1 (b), the obtained liveness-enforcing supervisor is the same with Fig. 5.1 (b). In this study, we design robust supervisors for S^3PR based on this divide-and-conquer deadlock control strategy.

5.3.2 Robustness of a Supervisor

Let $P_V = \{V_i | i \in \mathbb{N}_u\}$ be the set of the monitors computed by Algorithm 5.1 and $P_R = P_{R_1} \cup P_{R_2}$ be the set of the resource places, where $P_{R_1} = \{r | r \in R, M_0(r) = 1\}$ and $P_{R_2} = \{r | r \in R, M_0(r) \geq 2\}$. At a marking M, when a resource in p breaks down, we remove a token from p, i.e., $M(p) := M(p) - 1$. In order to analyze the robustness of a supervisor, we define a new type of states called waiting-for-repair states.

Definition 5.10 *Let $(N_V'^c, M_{V0}'^c)$ be a controlled system of an S^3PR net system (N', M_0'), where (N', M_0') is an augmented autonomous or an augmented toparchy subnet computed by Algorithm 5.1. Marking $M \in R(N_V'^c, M_{V0}'^c)$ is said to be a waiting-for-repair state if $\sum_{p \in \|I_r\|} M(p) <$*

$M_0(r)$, $\nexists t \in T'$ such that $M[t\rangle$, where $r \in P_R$.

Both deadlock and waiting-for-repair states are those at which an original system stops running. However, they have different generation mechanisms. At a deadlock state, a set of processes keeps waiting indefinitely for other processes in a set to release resources [35]. While at a waiting-for-repair state, a set of processes keeps waiting for resources that are removed to repair. At a waiting-for-repair state and deadlock, system stops running. For an S^3PR, there exists at least an SMS emptied at a deadlock state. As well known, a siphon remains empty once it is emptied. It means that the system will not run. While at a waiting-for-repair state, the system stops running and is waiting for some necessary resources. The system can work properly again when the repaired resources are returned. In essence, deadlocks are a safety property that is closely related to liveness, which can be avoided through proper control methods. While a waiting-for-repair state cannot be avoided since a resource failure is uncertain and stochastic.

For $r \in P_R$ in an S^3PR, the system must trap into a waiting-for-repair state when r fails with $M_0(r) = 1$ or when all this type of resources fails to work. This chapter discusses more complex and meaningful cases where a type of resources does not totally fail to work.

In an AMS, a resource failure is of temporal uncertainty. Correspondingly, unreliable resources may break down in activity or resource places in its Petri net model. $P_{U_r} = \{p_u | p_u \in \|I_r\|, r \in P_{R_2}\}$ is called the set of unreliable places of r. When a resource breaks down in an unreliable place p_u, we try to add a subnet that can remove a token from p_u and repair the broken resource. Also, after the resource is repaired, this subnet can return a token to the unreliable place. Then the resource can be used again. We call this subnet a recovery subnet. Its formal definition is proposed as follows.

Definition 5.11 *A recovery subnet is an ordinary Petri net $N_r = (\{p_u, p_{re}\}, \{t_s, t_f\}, F_r)$, where $F_r = \{(p_u, t_s), (t_s, p_{re}), (p_{re}, t_f), (t_f, p_u)\}$. (N_r, M_{r0}) is called a marked recovery subnet and $M_{r0} = x \cdot p_u$ is called an initial marking, where x is a nonnegative integer.*

Fig. 5.8 graphically represents a marked recovery subnet (N_r, M_{r0}). Place p_u represents an unreliable place in a plant model N. In this place, some failures may occur and be detected. Hence a recovery procedure is needed. Transitions t_s and t_f are called a start and a finish transition, respectively. They denote that the recovery activities start and finish. In fact, when a resource in p_u

fails to work, a recovery activity will initialize. The resource is moved away from p_u into p_{re} by firing t_s. Place p_{re} is called a recovery place of p_u. After the resource is repaired in p_{re}, it returns to p_u by firing t_f.

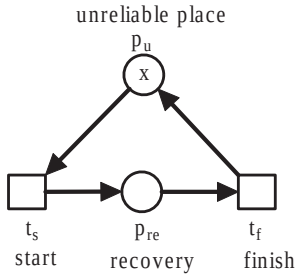


FIG. 5.8 – A recovery subnet.

Take Fig. 5.3 as an example. There are five recovery subnets. For resource type R, robots may break down in p_2, p_4, or p_6. Hence, $P_{U_R} = \{p_2, p_4, p_6\}$. When resources fail to work, a supervisor should start its corresponding recovery function.

From the running example in Section 3, we find that the removal of the broken resource influences the liveness of the supervisor. In other words, some monitors fail to control the corresponding SMS. Here, we first explore the relationship between broken resources and monitors.

Definition 5.12 *Let (N_V^c, M_{V0}^c) be a controlled system of an S^3PR, $r \in P_R$ be a resource, and $V \in P_V$ be a monitor. r is said to be correlated with monitor V if $H(r) \cap H(V) \neq \emptyset$. The set of correlated monitors of resource r is denoted by P_{V_r}.*

Definition 5.13 *Let (N_V^c, M_{V0}^c) be a controlled system of an S^3PR, V_S be the monitor of S, and $r \in P_R$ be a correlated resource to monitor $V_S \in P_{V_S}$. r and V_S are said to be strongly correlated if $r \in S$. Otherwise, they are said to be weakly correlated. The set of monitors strongly (weakly) correlated with resource r is denoted by $P_{V_{rs}}$ ($P_{V_{rw}}$). The number of monitors in $P_{V_{rs}}$ ($P_{V_{rw}}$) is denoted by θ (η), i.e., $\theta = |P_{V_{rs}}|$ ($\eta = |P_{V_{rw}}|$).*

Let (N_V^c, M_{V0}^c) be a controlled system of an S^3PR, V_S be the monitor of S, and P_{V_r} be the set of correlated monitors of resource r. A place $p \in H(r)$ and a monitor $V_S \in P_{V_r}$ have the following three potential relationships : $p \in [S]$, $p \in S$, and $p \in \mathcal{P}_S \setminus [S]$. Specifically, $P_{V_{rs}}^1 = \{V_{S_x} | p \in$

$H(r), p \in [S_x], p \in H(V_{S_x})\}$, $P^2_{V_{rs}} = \{V_{S_y} | p \in H(r), p \in S_y\}$, and $P^1_{V_{rw}} = \{V_{S_z} | p \in H(r), p \in \mathcal{P}_{S_z} \setminus [S_z], p \in H(V_{S_z})\}$. Let α, β, and γ denote $|P^1_{V_{rs}}|$, $|P^2_{V_{rs}}|$, and $|P^1_{V_{rw}}|$, respectively.

By Definition 5.13, a monitor in P_{V_r} is either strongly or weakly correlated with resource r. In other words, $P_{V_r} = P_{V_{rs}} \cup P_{V_{rw}}$ and $P_{V_{rs}} \cap P_{V_{rw}} = \emptyset$ hold. As shown in Fig. 5.1 (b), $H(p_5) = \{p_3\}$, $H(p_6) = \{p_2, p_4\}$, and $H(p_7) = \{p_2, p_3\}$. By $H(p_5) \cap H(p_7) = \{p_3\} \neq \emptyset$ and $H(p_6) \cap H(p_7) = \{p_2\} \neq \emptyset$, p_5 and p_6 are correlated with p_7. Also, $M_0(p_7)$ changes with $M_0(p_5)$ and $M_0(p_6)$. Hence, $P_{V_{p_5}} = P_{V_{p_{5s}}} = \{p_7\}$ and $P_{V_{p_6}} = P_{V_{p_{6s}}} = \{p_7\}$. Also, for p_2, we can obtain $P^1_{V_{rs}} = \{p_7\}$, $P^2_{V_{rs}} = \emptyset$, and $P^1_{V_{rw}} = \emptyset$; for p_4, we can obtain $P^2_{V_{rs}} = \{p_7\}$, $P^1_{V_{rs}} = \emptyset$, and $P^1_{V_{rw}} = \emptyset$.

As shown in Fig. 5.3, when a resource in p_6 fails to work, we need to remove it. Hence, a recovery subnet needs to be started. This means that t_7 or t_9 will be fired. In fact, when a resource in p_6 fails to work and monitor p_7 is marked, t_7 is then fired, otherwise t_9 is fired. In other words, repairing the broken resource in p_6 does not depend on whether p_7 is marked, but this activity may effect the number of tokens in p_7. Next, we introduce the complementary place of a k-bounded place p to implement these requirements.

Definition 5.14 *The complementary place of a k-bounded place p with initial marking $M_0(p) = k$ is a new place $\overline{p}$ satisfying ${}^\bullet\overline{p} = p^\bullet$, $\overline{p}^\bullet = {}^\bullet p$, where k is a nonnegative integer. The place $\overline{p}$ is initially unmarked.*

By construction, $\overline{p}$ is k-bounded. For each reachable marking M, $M(p) = k$ implies $M(\overline{p}) = 0$. This definition is important to understand the following algorithm that enhances the robustness of the liveness-enforcing supervisor computed by Algorithm 5.1 through adding recovery subnets, normal arcs, and the complementary places of monitors.

Algorithm 5.2

robust liveness-enforcing supervisor design

Input : $G = (N^c_V, M^c_{V0})$ *of an* S^3PR *plant model* (N, M_0)

Output : (N^{rc}_V, M^{rc}_{V0})

begin{

find P_{R_2}

while $(P_{R_2} \neq \emptyset)$ ***do***

choose a resource $r \in P_{R_2}$

$P_{R_2} := P_{R_2} \setminus \{r\}$

find P_{V_r}

if $P_{V_r} = \emptyset$ ***then***

add recovery subnets to every place in P_{U_r}

else

find $P_{V_{rs}}$, $P_{V_{rw}}$, $P^2_{V_{rs}}$, *and* $P^1_{V_{rw}}$

if $V_S \in P_{V_{rs}}$ *does not have a complementary place* $\overline{V_S}$ ***then***

add a complementary place $\overline{V_S}$ *for* V_S

end if

(i) for each $p \in P_{U_r} \setminus \{r\}$, *add* 2^β ($\beta = |P^2_{V_{rs}}|$) *recovery subnets to* p. *Either* $V_{rs} \in P^2_{V_{rs}}$ *or* $\overline{V_{rs}}$ *connects transition* t_{ps} *in every recovery subnet. A normal arc* (V_{rs}, t_{ps}) *is used to connect* V_{rs} *and* t_{ps}, *while a self-loop* $(\overline{V_{rs}}, t_{ps})$ *and* $(t_{ps}, \overline{V_{rs}})$ *weighted by* $M_{V0}(V_{rs})$ *connects* $\overline{V_{rs}}$ *and* t_{ps}. *There are* 2^β *logical combinations. If* V_{rs} *connects* t_{ps} *in a recovery subnet by a normal arc, in pairs, a normal arc from* t_{pf} *in the recovery subnet to* V_{rs} *is added. Correspondingly, add arcs* $(t_{ps}, \overline{V_{rs}})$ *and* $(\overline{V_{rs}}, t_{pf})$ *to ensure that* $\overline{V_{rs}}$ *is the complementary place of* V_{rs}. *Meanwhile, add arcs* (t_{ps}, V_{rw}) *and* (V_{rw}, t_{pf}) *for every recovery subnet, where* $V_{rw} \in P^1_{V_{rw}}$.

(ii) for resource r, *add* 2^θ ($\theta = |P_{V_{rs}}|$) *recovery-subnets. Either* $V_{rs} \in P_{V_{rs}}$ *or* $\overline{V_{rs}}$ *connects transition* t_{rs} *in every recovery subnet. A normal arc* (V_{rs}, t_{rs}) *is used to connect* V_{rs} *and* t_{rs}, *while a self-loop* $(\overline{V_{rs}}, t_{rs})$ *and* $(t_{rs}, \overline{V_{rs}})$ *weighted by* $M_{V0}(V_{rs})$ *connects* $\overline{V_{rs}}$ *and* t_{rs}. *There are* 2^θ *logical combinations. If* V_{rs} *connects* t_{rs} *in a recovery subnet by a normal arc, in pairs, a normal arc from* t_{rf} *of the recovery subnet to* V_{rs} *is added. Correspondingly, add arcs* $(t_{rs}, \overline{V_{rs}})$ *and* $(\overline{V_{rs}}, t_{rf})$ *to ensure that* $\overline{V_{rs}}$ *is the complementary place of* V_{rs}.

end if

end while

output a robust liveness-enforcing net (N^{rc}_V, M^{rc}_{V0})

}end of the algorithm

T_s and T_f are used to denote the sets of start and finish transitions in all recovery subnets, respectively. The algorithm indicates when some resources break down, we remove them to repair. At the same time we may need to change the number of tokens in their correlated monitors. If the

number of tokens in their correlated monitors is changed and broken resources are returned, the removed tokens in monitors will put back. Next results show that the liveness of the supervisor of the original S^3PR does not change and disturbed controlled system may be at waiting-for-repair states.

Theorem 5.5

(N_V^{rc}, M_{V0}^{rc}) synthesized by Algorithm 5.2 preserves the liveness of the supervisor designed by Algorithm 5.1 for an S^3PR.

Proof : Let us show that (N_V^{rc}, M_{V0}^{rc}) preserves the liveness of the supervisor designed by Algorithm 5.1 for the plant model (N, M_0). For $r \in P_{R_2}$, suppose that $H(r) = \{p_1, \ldots, p_l\}$ and $\|I_r\| = \{r, p_1, \ldots, p_l\}$ in (N, M_0).

1. $P_{V_r} = \emptyset$. In this case, r is not in any SMS in N. Algorithm 5.2 adds recovery subnets to every place in P_{U_r}, where $r \in P_{R_2}$. This implies that $\|I_r^{rc}\| = \{r, p_1, \ldots, p_l, p_{rr}, p_{1r}, \ldots, p_{lr}\}$ in (N_V^{rc}, M_{V0}^{rc}), where p_{rr}, p_{1r}, . . ., and p_{lr} are recovery places of r, p_1, . . ., and p_l, respectively. $\forall M \in R(N_V^{rc}, M_{V0}^{rc})$, $\exists t \in T \cup T_s \cup T_f$ such that $M[t\rangle$. Hence, adding recovery subnets to every place in P_{U_r} does not cause deadlocks.

2. $P_{V_r} \neq \emptyset$.

By Definition 5.14, adding complementary places does not effect the liveness of the whole supervisor.

(2.1) For each $p \in H(r)$: a place p and a monitor $V_S \in P_{V_r}$ have the following three potential relationships : $p \in [S]$, $p \in S$, and $p \in \mathcal{P}_S \setminus [S]$. Specifically, we have $P^1_{V_{rs}} = \{V_{S_x} | p \in H(r), p \in [S_x], p \in H(V_{S_x})\}$, $P^2_{V_{rs}} = \{V_{S_y} | p \in H(r), p \in S_y\}$, and $P^1_{V_{rw}} = \{V_{S_z} | p \in H(r), p \in \mathcal{P}_{S_z} \setminus [S_z], p \in H(V_{S_z})\}$. α, β, and γ are used to denote $|P^1_{V_{rs}}|$, $|P^2_{V_{rs}}|$, and $|P^1_{V_{rw}}|$, respectively. In other words, p is in the complementary sets of α SMS, in β SMS, and in $\mathcal{P}_{S_z} \setminus [S_z]$ of γ SMS.

(a) p is in the complementary sets of α SMS : if the resource in p breaks down, the removal of the broken resource means that the token count in their monitors decreases by one implicitly. Hence, these monitors in $P^1_{V_{rs}}$ act normally to control their corresponding SMS.

(b) p is in β SMS : to keep the liveness of the supervisor, certain mechanism needs to be designed between p and the corresponding β monitors. This mechanism should satisfy two basic requirements : on the one hand, when the resource represented by p breaks down, the removal

of the resource effects the token count of their monitors that are marked. On the other hand, the removal of the broken resource does not depend on whether these β corresponding monitors are marked. For β monitors, there are 2^β logical combinations to indicate which one is marked and which one is unmarked. Algorithm 5.2 uses the complementary places of these monitors to show that when the monitors are unmarked, the start transitions can also be fired.

Algorithm 5.2 adds 2^β recovery subnets to p. Indeed, when a resource in p fails at some markings, only one of these 2^β recovery subnets will be started next. The fact which recovery subnet will work is based on the markings. In Algorithm 5.2, normal arcs (V_{rs}, t_{ps}) and (t_{ps}, V_{rs}) are used to connect V_{rs}, t_{ps}, and t_{pf}. This implies that at those markings, if V_{rs} is marked, arc (V_{rs}, t_{ps}) is enabled. Correspondingly, add arcs $(t_{ps}, \overline{V_{rs}})$ and $(\overline{V_{rs}}, t_{pf})$ to ensure that $\overline{V_{rs}}$ is the complementary place of V_{rs}. A self-loop $(\overline{V_{rs}}, t_{ps})$ and $(t_{ps}, \overline{V_{rs}})$ weighted by $M_{V0}(V_{rs})$ is added to connect $\overline{V_{rs}}$ and t_{ps}. This implies that at those markings, if V_{rs} is unmarked, then $\overline{V_{rs}}$ is marked by $M_{V0}(V_{rs})$ and arc $(\overline{V_{rs}}, t_{ps})$ is enabled.

(c) p is in $\mathcal{P}_{S_z} \setminus [S_z]$ of γ SMS : for $V_{rw} \in P^1_{V_{rw}}$, the token count in V_{rw} should not change with that of p. Hence, adding arcs (t_{ps}, V_{rw}) and (V_{rw}, t_{pf}) can ensure that the token count in V_{rw} designed for S_z does not change.

(2.2) Similarly, step (ii) in Algorithm 5.2 ensures that the token counts of the strongly correlated monitors of r change with that in r on condition that monitors can still control their corresponding siphons.

In a word, (N^{rc}_V, M^{rc}_{V0}) designed by Algorithm 5.1 for the plant model (N, M_0) preserves the liveness of the supervisor. □

Theorem 5.6

Let $(N'^{rc}_V, M'^{rc}_{V0})$ be a controlled system of an S^3PR (N', M'_0), where (N', M'_0) is an augmented autonomous or an augmented toparchy subnet computed by Algorithm 5.1. $(N'^{rc}_V, M'^{rc}_{V0})$ with broken resources is at a waiting-for-repair state if $\exists M \in R(N'^{rc}_V, M'^{rc}_{V0})$, $\sum_{p\in\|I_r\|} M(p) < M_0(r)$ and $\nexists t \in T'$, $M[t\rangle$, where $r \in P_R$.

Proof : For $(N'^{rc}_V, M'^{rc}_{V0})$, if some resources break down and are removed to repair, the disturbed system stops running and is waiting for the removed resources to be returned. We have two cases.

1. $P_{V_r} = \emptyset$. Algorithm 5.2 adds recovery subnets to every place in P_{U_r}, where $r \in P_{R_2}$. When $M_0(r)$ resources break down and all of them are removed to recovery subnets and meanwhile the system needs this resource type, the system is at a waiting-for-repair state.

2. $P_{V_r} \neq \emptyset$. Adding arcs (t_{ps}, V_{rw}) and (V_{rw}, t_{pf}) does not change token count of V_{rw}, where $V_{rw} \in P^1_{V_{rw}}$ in Algorithm 5.2. However, other steps in (i) and (ii) in Algorithm 5.2 force broken resources to be removed to recovery subnets. From Definition 5.10, if $\exists M \in R(N'^{rc}_V, M'^{rc}_{V0})$, $\sum_{p \in \|I_r\|} M(p) < M_0(r)$ and $\nexists t \in T'$, $M[t\rangle$, the disturbed controlled system is at a waiting-for-repair state, where $r \in P_R$. □

Algorithm 5.2 is applied to the net depicted in Fig. 5.1 (b). Since $P_{R_2} = \{p_6\}$, it needs to iterate only once. We have $P_{V_{p_{6s}}} = \{p_7\}$ and $P_{V_{p_{6w}}} = \emptyset$. The complementary place $\overline{p_7}$ needs to be added for p_7. For p_2 that is in the complementary set of SMS $\{p_4, p_5, p_6\}$, $P^2_{V_{p_{6s}}} = \emptyset$ implies that $\beta = 0$ and a recovery subnet is needed. For p_4 that is in SMS $\{p_4, p_5, p_6\}$, $P^2_{V_{p_{6s}}} = \{p_7\}$ implies that $\beta = 1$ and two recovery subnets are needed. As shown in Fig. 5.3, normal arcs (p_7, t_{11}) and (t_{12}, p_7) are used to connect the monitor with a recovery subnet, and a self-loop $(\overline{p_7}, t_{13})$ and $(t_{13}, \overline{p_7})$ weighted by 2 is used to connect $\overline{p_7}$ and the other recovery subnet. For resource place r, $\theta = |P_{V_{p_{6s}}}| = 1$. Hence two recovery subnets need to be added. Normal arcs (p_7, t_7) and (t_8, p_7) are used to connect the monitor with a recovery subnet, and a self-loop $(\overline{p_7}, t_9)$ and $(t_9, \overline{p_7})$ weighted by 2 is used to connect $\overline{p_7}$ and the other recovery subnet. Then (N^{rc}_V, M^{rc}_{0V}) is obtained for the original S^{3}PR, as shown in Fig. 5.3.

Based on Theorem 5.6, the final system obtained by Algorithm 5.2 can trap into waiting-for-repair states. For example, in Fig. 5.3, when two robots are occupied by p_2 and p_4 and the robot in place p_4 breaks down, it needs to be repaired. However, after removing this robot to p_{12}, we find that the original system stops running. It gets into a waiting-for-repair state. Note that (N^{rc}_V, M^{rc}_{0V}) is still live.

Indeed, resource failures are of randomness. Specifically, the resources that will fail, the states at which resources will fail, and the number of broken resources are unpredicted before errors occur. The objective of this chapter is to design a robust supervisor that can handle errors as many as possible without reanalyzing the original system. However, based on Algorithm 5.2, the robust liveness-enforcing supervisor for an S^{3}PR has a complex structure. The next section shows that the supervisor structure can be reduced by introducing inhibitor arcs.

5.3.3 Improvement of Algorithm 5.2

The class of Petri nets studied in this chapter are bounded ones. A bounded Petri net with inhibitor arcs can be transformed into a conventional Petri net [22]. Given a k-bounded place p with ${}^\bullet p = T^* = \{t_1, \cdots, t_m\}$, $p^\bullet = T^{*\prime} = \{t'_1, \cdots, t'_n\}$, and $p^{o\bullet} = \{t\}$, the behavior of inhibitor arc $(p,t)^o$ can be equivalently replaced by adding a complementary place $\overline{p}$ with ${}^\bullet\overline{p} = T^{*\prime} = \{t'_1, \cdots, t'_n\}$, $\overline{p}^\bullet = T^* = \{t_1, \cdots, t_m\}$ for p and two normal arcs from and to $\overline{p}$, weighted by k. For each reachable marking, p is unmarked if and only if $\overline{p}$ carries k tokens. Then the inhibitor arc is replaced. Fig. 5.9 illustrates this equivalent transformation. This property of inhibitor arcs in bounded Petri nets is significant for us to simplify the supervisor structure derived from Algorithm 5.2.

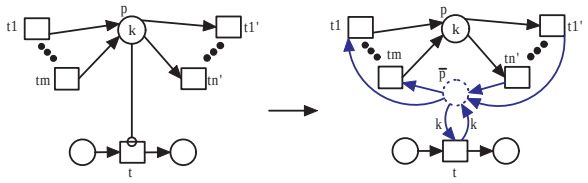


FIG. 5.9 – Transformation.

In Algorithm 5.2, the complementary places of monitors and related arcs are used to represent the fact that the removal of the broken resources does not depend on the markings of their related monitors. Specifically, if a resource fails in SMS and is removed and the corresponding monitors are marked, a token from them is removed at the same time. If the corresponding monitors are unmarked, it does not effect the removal of the broken resource. When a monitor V_S loses $M_{0V}(V_S)$ tokens, $\overline{V_S}$ must get $M_{0V}(V_S)$ tokens. Then the corresponding start transition can be fired. Indeed, an inhibitor arc can be used to represent the fact that when the corresponding monitors are unmarked, it does not effect the removal of the broken resource. Inhibitor arcs are used to replace the complementary places of monitors and related arcs. Algorithm 5.2 can be improved accordingly.

Algorithm 5.3

robust liveness-enforcing supervisor design with inhibitor arcs

Input : $G = (N_V^c, M_{V0}^c)$ *of an* S^3PR *plant model* (N, M_0)

Output : $(N_V^{rc*}, M_{V0}^{rc*})$

begin{

find P_{R_2}

while *(*$P_{R_2} \neq \emptyset$*)* ***do***

choose a resource $r \in P_{R_2}$

$P_{R_2} := P_{R_2} \setminus \{r\}$

find P_{V_r}

if $P_{V_r} = \emptyset$ ***then***

add recovery subnets to every place in P_{U_r}

else

find $P_{V_{rs}}$, $P^2_{V_{rs}}$, *and* $P^1_{V_{rw}}$

(i) for each $p \in P_{U_r} \setminus \{r\}$, *add* 2^β ($\beta = |P^2_{V_{rs}}|$) *recovery subnets to* p. *Each* $V_{rs} \in P^2_{V_{rs}}$ *connects transition* t_{ps} *in every recovery subnet by a normal or inhibitor arc. There are* 2^β *logical combinations. If* V_{rs} *connects* t_{ps} *in a recovery subnet by a normal arc, in pairs, a normal arc from* t_{pf} *in the recovery subnet to* V_{rs} *is added. Meanwhile, add arcs* (t_{ps}, V_{rw}) *and* (V_{rw}, t_{pf}) *for every recovery subnet, where* $V_{rw} \in P^1_{V_{rw}}$.

(ii) for resource r, *add* 2^θ ($\theta = |P_{V_{rs}}|$) *recovery-subnets. Each* $V_{rs} \in P_{V_{rs}}$ *connects* t_{rs} *in every recovery subnet by a normal or inhibitor arc. There are* 2^θ *logical combinations. If* V_{rs} *connects* t_{rs} *in a recovery subnet by a normal arc, in pairs, a normal arc from* t_{rf} *in the recovery subnet to* V_{rs} *is added.*

end if

end while

output a robust liveness-enforcing net $(N^{rc*}_V, M^{rc*}_{V0})$

}end of the algorithm

$(N^{rc*}_V, M^{rc*}_{V0})$ is a Petri net with inhibitor arcs. It is easy to find that the structure of $(N^{rc*}_V, M^{rc*}_{V0})$ is more simple than that of (N^{rc}_V, M^{rc}_{V0}). The complementary place of a monitor and a self-loop between the complementary place and a start transition are equivalent with an inhibitor arc from the monitor to the start transition. Algorithm 5.3 is equivalent with Algorithm 5.2 on behavior property. Hence, Theorems 5.7 and 5.8 hold, whose proofs are omitted.

Theorem 5.7

$(N^{rc*}_V, M^{rc*}_{V0})$ *synthesized by Algorithm 5.3 preserves the liveness of the supervisor designed by Algorithm 5.1 for an* S^3PR.

Theorem 5.8

Let $(N_V'^{rc*}, M_{V0}'^{rc*})$ *be a controlled system of an* S^3PR (N', M_0')*, where* (N', M_0') *is an augmented autonomous or an augmented toparchy subnet computed by Algorithm 5.1.* $(N_V'^{rc*}, M_{V0}'^{rc*})$ *with broken resources is at a waiting-for-repair state if* $\exists M \in R(N_V'^{rc*}, M_{V0}'^{rc*})$, $\sum_{p\in\|I_r\|} M(p) < M_0(r)$ *and* $\nexists t \in T'$, $M[t\rangle$, *where* $r \in P_R$.

Algorithm 5.3 is applied to the net depicted in Fig. 5.1 (c). Since $P_{R_2} = \{p_6\}$, it needs to iterate only once. We have $P_{V_{p_{6s}}} = \{p_7\}$ and $P_{V_{p_{6w}}} = \emptyset$. For p_2 that is in the complementary set of SMS $\{p_4, p_5, p_6\}$, $P^2_{V_{p_{6s}}} = \emptyset$ implies $\beta = 0$ and a recovery subnet is needed. For p_4 that is in SMS $\{p_4, p_5, p_6\}$, $P^2_{V_{p_{6s}}} = \{p_7\}$ implies $\beta = 1$ and two recovery subnets need to be added. As shown in Fig. 5.10, normal arcs (p_7, t_{11}) and (t_{12}, p_7) and inhibitor arc $(p_7, t_{13})^o$ are used to connect the monitor with recovery subnets. For resource place r, $\theta = |P_{V_{p_{6s}}}| = 1$. Hence two recovery subnets need to be added. Normal arcs (p_7, t_7) and (t_8, p_7) and inhibitor arc $(p_7, t_9)^o$ are used to connect monitor with recovery subnets. Then we have $(N_V^{rc*}, M_{0V}^{rc*})$ for the original S^3PR, as shown in Fig. 5.10.

Based on Theorem 5.8, in Fig. 5.10, when two robots are occupied by p_2 and p_4 and the robot in place p_4 breaks down, the robot needs to be repaired. After removing this robot to p_{12}, we find that the system stops running. It gets into a waiting-for-repair state. However, $(N_V^{rc*}, M_{0V}^{rc*})$ is still live.

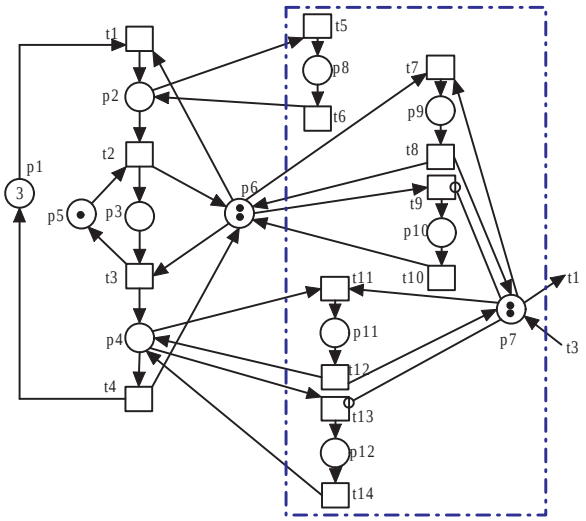


FIG. 5.10 – Supervisor with inhibitor arcs.

Compared Figs. 5.10 with 5.3, complementary place $\overline{p}$ and related arcs are replaced by an inhibitor arc. The structure of the robust liveness-enforcing supervisor is simplified.

5.4 Examples

The automated manufacturing cell shown in Fig. 5.11(a) has three types of machines M1, M2, and M3. M2 has two processing units while M1 and M3 have only one unit. Also the cell contains two types of robots R1 and R2 and each type has one processing unit. Parts enter the cell through two loading buffers I1 and I2, and leave the cell through two unloading buffers O1 and O2. The robots deal with the movements of parts. R1 handles part movements from I1 to M1 and M2. R2 handles part movements from M2 to M3 and M3 to M2. Three part types J1, J2, and J3 are produced. Their respective production routes are shown in Fig. 5.11(b) and the Petri net model of the system is shown in Fig. 5.4(a). The net system is an S^3PR that contains deadlocks. For the net in Fig. 5.4(a), a liveness-enforcing monarch (N_V^c, M_{0V}^c) synthesized by Algorithm 5.1 is shown in Fig. 5.7.

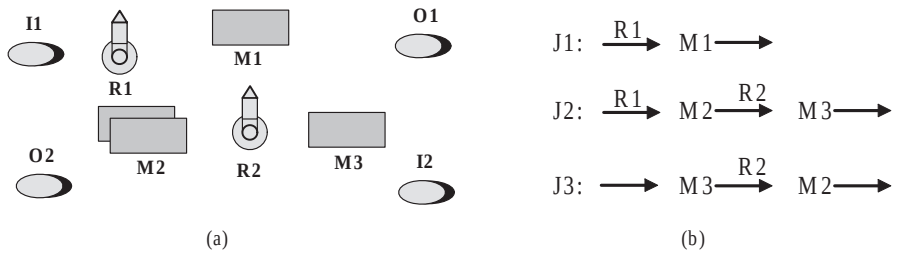


FIG. 5.11 – (a) An AMS's layout, (b) the production routings of the AMS.

In the original S^3PR model, there are two tokens in p_{12} (M_2), i.e., $P_{R_2} = \{p_{12}\}$. The method proposed in this chapter is applicable. Specifically, from the production routes, we observe : (1) if R_1 breaks down when R_1 moves J_1 and J_2, J_1 and J_2 cannot be successfully produced ; (2) if M_1 breaks down, J_1 cannot be finished ; (3) if R_2 or M_3 fails to work, J_2 and J_3 cannot be finished ; and (4) if one machine of M_2 fails to work and the other one works properly at some states, system may produce parts smoothly without reanalyzing by Algorithm 5.3.

There are three SMS $S_1 = \{p_5, p_9, p_{12}, p_{13}\}$, $S_2 = \{p_4, p_6, p_{13}, p_{14}\}$, and $S_3 = \{p_6, p_9, p_{12}, p_{13}, p_{14}\}$. Their complementary sets are $[S_1] = \{p_3, p_4\}$, $[S_2] = \{p_5, p_8\}$, and $[S_3] = \{p_3, p_4, p_5, p_8\}$,

respectively. By Definitions 5.12 and 5.13, we find $P_{V_{p_{12}}} = \{V_{S_1}, V_{S_2}, V_{S_3}\}$, $P_{V_{p_{12s}}} = \{V_{S_1}, V_{S_3}\}$, and $P_{V_{p_{12w}}} = \{V_{S_2}\}$. For resource type p_{12}, $H(p_{12}) = \{p_3, p_9, p_{12}\}$. As shown in Fig. 5.7, for p_3, we can obtain $\alpha = |\{V_{S_x} | p \in H(r), p \in [S_x], p \in H(V_{S_x})\}| = |\{V_{S_1}, V_{S_3}\}| = 2, \beta = |\{V_{S_y} | p \in H(r), p \in S_y\}| = 0$, and $\gamma = |\{V_{S_z} | p \in H(r), p \in \mathcal{P}_{S_z} \setminus [S_z], p \in H(V_{S_z})\}| = |\{V_{S_2}\}| = 1$. By Algorithm 5.3, a recovery subnet and arcs (t_{20}, V_{S_2}) and (V_{S_2}, t_{21}) are needed to add for p_3 as shown in Fig. 5.12. The token count in V_{S_2} is compensated by these two arcs.

For p_9, we can obtain $\alpha = |\{V_{S_x} | p \in H(r), p \in [S_x], p \in H(V_{S_x})\}| = |\emptyset| = 0, \beta = |\{V_{S_y} | p \in H(r), p \in S_y\}| = |\{V_{S_1}, V_{S_3}\}| = 2$, and $\gamma = |\{V_{S_z} | p \in H(r), p \in \mathcal{P}_{S_z} \setminus [S_z], p \in H(V_{S_z})\}| = |\emptyset| = 0$. According to Algorithm 5.3, four recovery subnets are needed to add for p_9. If resource in p_9 breaks down, it needs to be removed, which does not depend on whether V_{S_1} and V_{S_3} are marked. According to this fact, normal and inhibitor arcs are added for monitors and recovery subnets as shown in Fig. 5.12.

Since $|P_{V_{p_{12s}}}| = |\{V_{S_1}, V_{S_3}\}| = 2$, we also need to add four recovery subnets to p_{12}. If the resource in p_{12} breaks down, it needs to be removed, which does not depend on whether V_{S_1} and V_{S_3} are marked. Accordingly, normal and inhibitor arcs are added for monitors and recovery subnets as Fig. 5.12 shows. Then a robust liveness-enforcing supervisor $(N_V^{rc*}, M_{0V}^{rc*})$ for the original S^{3}PR is derived.

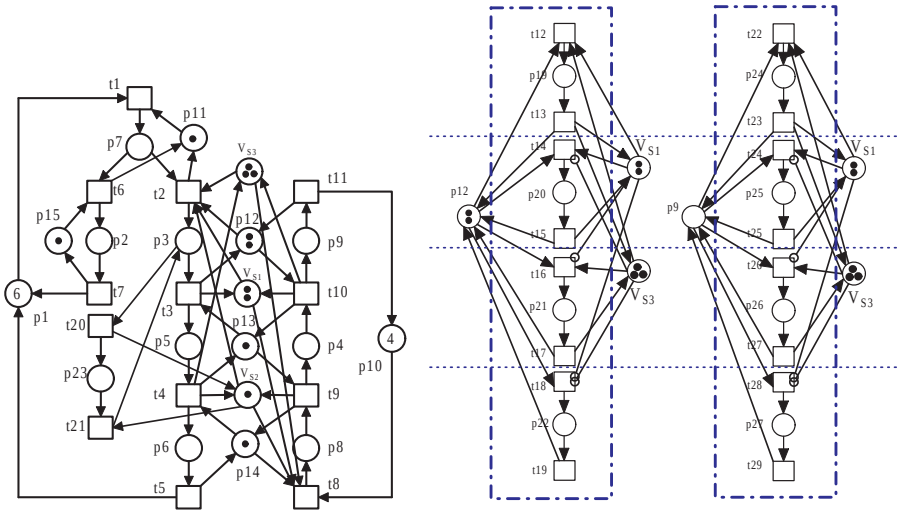


FIG. 5.12 – A robust supervisor for the original system.

In the above AMS example, we consider only a type of resources is unreliable. Next we show a

more complex example with four unreliable resource types, one of which does not have correlated monitors. The Petri net model shown in Fig. 5.13 is an S^{3}PR, where $P^0 = \{p_1, p_{11}, p_{18}, p_{33}\}$, $P_R = \{p_{12}, p_{13}, p_{14}, p_{15}, p_{16}, p_{17}, p_{24}, p_{25}, p_{26}, p_{34}, p_{35}, p_{36}\}$, and the others are activity places.

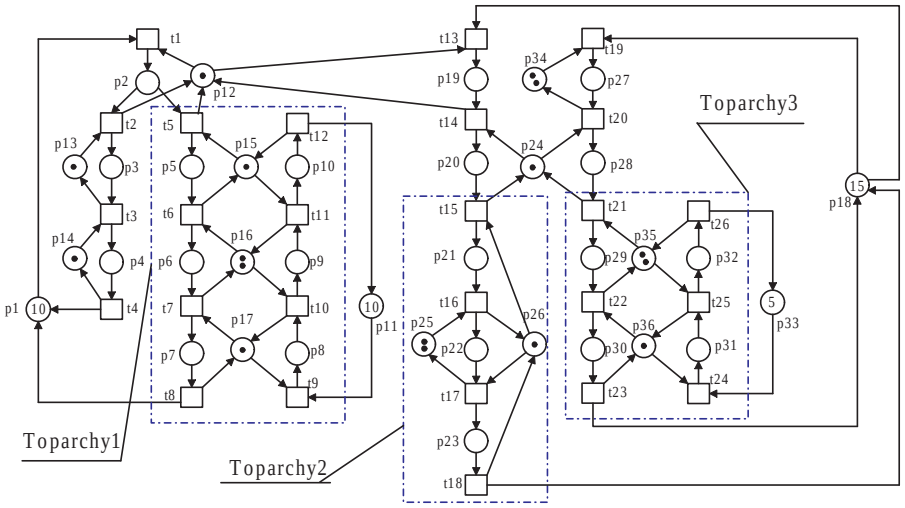


FIG. 5.13 – Plant model (N, M_0).

As discussed in Section 4.1.2, we have $P_R^F = \{p_{12}, p_{13}, p_{14}, p_{24}, p_{34}\}$, $P_R^1 = \{p_{15}, p_{16}, p_{17}\}$, $P_R^2 = \{p_{25}, p_{26}\}$, and $P_R^3 = \{p_{35}, p_{36}\}$. This resource place partition leads to an idle subnet, an autonomous subnet, and three toparchies that are boxed in rectangles with dash lines, as depicted in Fig. 5.13. Toparchy 1 has three SMS : $S_1 = \{p_6, p_{10}, p_{15}, p_{16}\}$, $S_2 = \{p_7, p_9, p_{16}, p_{17}\}$, and $S_3 = \{p_7, p_{10}, p_{15}, p_{16}, p_{17}\}$. Toparchy 2 has one SMS $S_4 = \{p_{23}, p_{25}, p_{26}\}$ and Toparchy 3 has one SMS $S_5 = \{p_{30}, p_{32}, p_{35}, p_{36}\}$. By the deadlock prevention policy proposed in Definition 5.2, each SMS can be controlled by adding a monitor such that an S^{3}PR is live. Accordingly, for each of toparchies 1, 2, and 3, a toparch can be designed by synthesizing a set of monitors, as shown in Table 5.1. Once these toparches are computed, a monarch can be found by synchronously synthesizing the idle subnet, the autonomous subnet, and the toparches.

In the original S^{3}PR model, $P_{R_2} = \{r | r \in R, M_0(r) \geq 2\} = \{p_{16}, p_{25}, p_{34}, p_{35}\}$. We first choose unreliable resource p_{16} that is in Toparchy 1 as an example. By Definitions 5.12 and 5.13, $P_{V_{p_{16}}} = \{V_{S_1}, V_{S_2}, V_{S_3}\}$, $P_{V_{p_{12s}}} = \{V_{S_1}, V_{S_2}, V_{S_3}\}$, and $P_{V_{p_{12w}}} = \emptyset$. For resource type p_{16}, $H(p_{16}) = \{p_6, p_9, p_{16}\}$. For p_6, we can obtain $\alpha = |\{V_{S_x} | p \in H(r), p \in [S_x], p \in H(V_{S_x})\}| = |\{V_{S_2}, V_{S_3}\}| = 2$, $\beta = |\{V_{S_y} | p \in H(r), p \in S_y\}| = \{V_{S_1}\} = 1$, and $\gamma = |\{V_{S_z} | p \in H(r), p \in \mathcal{P}_{S_z} \setminus [S_z], p \in H(V_{S_z})\}| =$

Tab. 5.1 – Monitors in the toparches for three toparchies

toparch	monitor	preset	postset	marking
Toparch 1	V_{S_1}	t_6, t_{11}	t_5, t_9	2
	V_{S_2}	t_7, t_{10}	t_5, t_9	2
	V_{S_3}	t_7, t_{11}	t_5, t_9	3
Toparch 2	V_{S_4}	t_{17}	t_{15}	2
Toparch 3	V_{S_5}	t_{22}, t_{25}	t_{21}, t_{24}	2

$|\emptyset| = 0$. According to Algorithm 5.3, two recovery subnets are needed to add for p_6 and normal and inhibitor arcs are needed to connect monitor V_{S_1} with these two subnets. Similarly, we can add recovery subnets and corresponding arcs for p_9 as shown in Fig. 5.14.

Since $|P_{V_{p_{16s}}}| = |\{V_{S_1}, V_{S_2}, V_{S_3}\}| = 3$, we need to add 8 recovery subnets to p_{16}. If the resource in p_{16} breaks down, it needs to be removed, which does not depend on whether V_{S_1}, V_{S_2}, and V_{S_3} are marked. Accordingly, normal and inhibitor arcs are added for monitors and recovery subnets as Fig. 5.14 shows. Similarly, we can design recovery subnets for p_{25}, p_{34}, and p_{35}. Then we obtain a robust liveness-enforcing supervisor $(N_V^{rc*}, M_{0V}^{rc*})$ for the original S^{3}PR as shown in Fig. 5.14. Due to the limited space, the supervisor is not shown completely. In the figure, for clarity, Ei is used to denote a recovery mechanism for an unreliable place, where $i = \{1, 2, \cdots, 10\}$. It is explained below the figure.

5.5 Discussions

There is a lack of research regarding unreliable resources on AMS under the existing deadlock control policies. The goal of this chapter is to design a robust liveness-enforcing supervisor in a Petri net formalism for AMS.

Based on the concept of resource circuits, we divide the resources in P_R of a considered S^{3}PR into two classes : the ones each of which is associated with a resource circuit from which an SMS can be derived from, and the ones that are not associated with SMS. The former is in $P_R^1 \cup P_R^2 \cup \cdots \cup P_R^k$ and the latter is in P_R^F, i.e., $P_R^1 \cup P_R^2 \cup \cdots \cup P_R^k \cup P_R^F = P_R$. According to this partition, an S^{3}PR is disassembled into an autonomous subnet if it exists, a number of toparchies, and an idle subnet. Then, a liveness-enforcing toparch is designed for each toparchy. All toparches and the

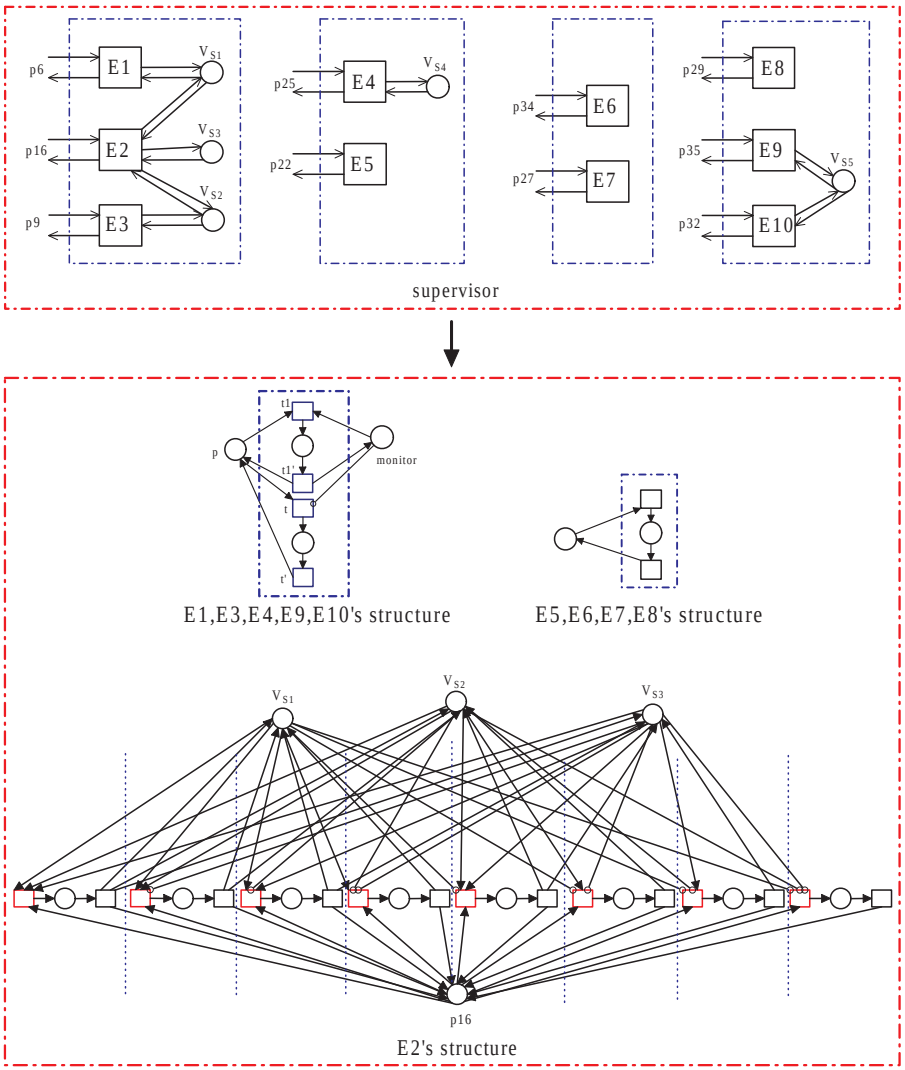


FIG. 5.14 – A robust supervisor for the original system.

autonomous subnet are merged into a monarch that is shown to be a liveness-enforcing supervisor for the whole plant Petri net model. We add recovery subnets to the unreliable places of unreliable resources. To keep the robustness of the monitors, normal and inhibitor arcs are needed to connect monitors and recovery subnets by Algorithm 5.3. Then, we can obtain a more robust supervisor than [76]. Note that when a waiting-for-repair state appears, the whole system must wait for the necessary resource that is removed to repair.

The chapter bridges the gap between the existing deadlock policies and real-world manufacturing systems. It provides deep insights on the development of deadlock control policies for AMS with unreliable resources. Compared with existing policies [23], [16], [89], [46], [104], [90], [115], [116], [117], [69], [56], [123], [13], we consider the uncertainty factors in AMS. By considering Hsieh's work [39], [40], [41], [42], [43], [44], [45], our research is more intuitive since supervisors are in a Petri net formalism.

We are led to conclude that the robustness design methodology developed in this research can be easily extended to other deadlock control policies based on siphons [16], [89], [69]. The reason why the divide-and-conquer deadlock control policy is selected to be a demonstrator rather than the typical method in [23] lies in the following aspects. First, supervisors designed by the method in [23] are expressed by a set of monitors that are added for emptiable SMS with output arcs pointing to the source transitions of the plant models. This usually forbids a portion of legal states. While the divide-and-conquer deadlock control policy can solve this problem to some extent since the allocation mechanism of the resources in P_R^F in a plant S^3PR net model cannot lead to deadlocks. Second, if we use the method in [23] at the first stage directly, by Definition 5.12, P_{V_r} of r may have more elements. This leads to the supervisor computed by Algorithm 5.3 with a much more complex structure. Last but not least, we will show why toparchies are not distinguished into subordinate or dominate [76] toparchies in Algorithm 5.1. Taking no account of unreliable resources, a subordinate toparchy's idle-augmented net is live based on its definition. However, if there exist unreliable resources in a subordinate toparchy and when some fail to work and are removed to repair, the subordinate toparchy's idle-augmented net may no longer live. In this case, monitors are needed to add to control deadlocks. Hence, at the first stage, monitors are added for all toparchies.

This chapter presents exploratory research on robustness of liveness-enforcing supervisors for

AMS in a Petri net formalism. Computational and structural complexity of the supervisors needs to be improved. More specifically, deciding how to reduce the complexity of Algorithm 5.3 and to extend the results to more complex Petri nets such as GS^3PR and S^4R [80] is of importance and significance. The robustness analysis problem of more general Petri nets remains open.

5.6 Summary

This chapter focuses on the robust liveness-enforcing supervisor design for AMS. Recently, extensive work on supervisor design based on Petri nets has been devoted to the liveness enforcement on the premise that all the resources in a system work properly. However, real-world AMS may suffer from unpredicted resource failures. When broken resources are removed to repair, the liveness-enforcing supervisor designed for the original system by traditional policies may cause deadlocks. This chapter is motivated by the need to design a supervisor that can ensure not only the liveness of the original Petri net but also the robustness of the controlled system for S^3PR.

Recovery subnets and monitors are added for unreliable resources and strict minimal siphons that may be emptied, respectively. Normal and inhibitor arcs are used to connect monitors with recovery subnets when necessary. In this case, when a resource fails, the proposed supervisor can still ensure that no deadlock occurs. Compared with the traditional deadlock control policies, the most advantage of the proposed method is that the reanalysis of the net can be avoided. To a large extent, the robustness of the supervisor is improved. Results indicate that the controller is qualified with robustness and liveness. However, there is an obvious drawback in this study. The final supervisor for S^3PR designed by Algorithm 5.3 is too complex in structure even though the algorithm is easily implemented for an S^3PR in theory.

Future work includes simplifying the structure of the supervisor designed by Algorithm 5.3 for S^3PR and extending the proposed technique to more general Petri nets such as GS^3PR and S^4R [80].

The major contribution in this research is published :

[1] **Gaiyun Liu**, Zhiwu Li, Kamel Barkaoui, and Abdulrahman M. Al-Ahmari, Robustness of deadlock control for a class of Petri nets with unreliable resources, *Information Sciences*, vol.235, pp.259-279, 2013.

Conclusions and Future Research

This chapter concludes this thesis by reviewing the major contributions, discussing the limitations of the proposed methods, and summarizing some future research topics.

1. Contributions

This thesis investigates some important deadlock control issues in automated manufacturing systems (AMS) based on Petri nets structural analysis and robust supervisor design. Three challenges are tackled. The first is the derivation of looser controllability condition of siphons. The second is the design of supervisors with near-optimal/optimal permissive behavior and low computational complexity. The third is the robust liveness-enforcing supervisor design for AMS with unreliable resources. The first challenge is addressed by proposing the concept of max*-controllability and integer programming (IP) test techniques. The second problem is approached by combining of the theory of regions and structural analysis. Further, a maximally permissive control policy for a subclass of Petri nets based on the theory of token distribution pattern of siphons is proposed. There is no need to construct a reachability graph and enumerate all minimal siphons. The third issue is overcome by introducing recovery subnets to Petri net plants. A robust liveness-enforcing supervisor is designed such that a good trade-off between the existing Petri net control policies and their application to real-world systems with unreliable resources can be made. The contributions of this thesis can be summarized into five aspects.

Firstly, it reviews the concepts of max, max$'$, and max$''$-controlled siphons and formulates the new concept called max*-controlled siphons for GS^{3}PR. We conclude that a GS^{3}PR is live iff all its siphons are max*-controlled. Then examples are given to illustrate the max, max$'$, max$''$, max*-controlled siphons and their difference. Compared with the existing work, the proposed me-

thod is more general. Also, some open problems are discussed. Based on the max*-controllability condition of siphons, for GS^3PR, it proposes a new IP model that can detect minimal problematic siphons directly. We conclude that if there is no feasible solution to this model, the net is live. Since the approach is based on siphons and mathematical programming, its computational efficiency is relatively insensitive to the initial marking. Compared with the existing methods, the proposed one is more powerful.

Secondly, it develops a novel design method of deadlock prevention supervisors based on Petri nets, which does not guarantee optimality but empirical results show its superiority over other approaches based on siphon control. Given the Petri net model of an AMS, an optimal liveness-enforcing controlled system is designed for the model under a minimal initial marking by utilizing the theory of regions. Then, we calculate all strict minimal siphons (SMS) in the controlled system, each of which does not contain a trap. For each SMS, an algebraic inequality with respect to the markings of monitors and resource places in the controlled system, also called a liveness constraint, is established in terms of the concept of max-controlled or invariant-controlled siphons. Its satisfaction implies the absence of dead transitions in the postset of the corresponding siphon. Consequently, given initial markings that satisfy all the liveness inequality constraints, all siphons can be max-controlled, and the resulting controlled system is live. After a controlled system structure is found, one can reallocate the initial markings according to the inequality constraints. No matter how large the initial markings and the number of states are, the liveness constraints remain unchanged. Their satisfaction ensures the absence of uncontrolled siphons.

Moreover, it proposes a maximally permissive control policy for a subclass of S^3PR (called β-nets) based on the theory of token distribution pattern of siphons. We first show that by adding a monitor for each critical siphon, some live states may get lost since the monitor is associated with the complementary set of a critical siphon, where some places may not be marked. By controlling only the set of marked activity places, more live states can be reached. However, this induces some emptiable siphons. The corresponding token pattern can be inferred. By adding monitors to all such possibly emptiable siphons, the controlled net becomes live and maximally permissive. There is no need to construct a reachability graph and enumerate all minimal siphons. Hence, the computational burden is minimized among all approaches in the literature.

Last but not least, a variety of deadlock control policies based on Petri nets have been pro-

posed for AMS. Most of them prevent deadlocks by adding monitors for emptiable siphons that, without an appropriate control policy, can cause deadlocks, where the resources in a system under consideration are assumed to be reliable. When resources are unreliable, it is difficult or impossible to apply existing control strategies. For S^3PR, Chapter 5 bridges the gap between a divide-and-conquer deadlock control strategy and its application to real-world systems with unreliable resources. Recovery subnets and monitors are designed for unreliable resources and strict minimal siphons that may be emptied, respectively. Normal and inhibitor arcs are used to connect monitors with recovery subnets if necessary. Then reanalysis of the original Petri net is avoided and a robust liveness-enforcing supervisor is derived.

2. Limitations and Future Research

Despite some basic problems that have been discussed and solved in this thesis, from many aspects, the work in the thesis can be further extended in the future. Such extensions making the proposed methods more practical and leading to more applications are discussed as follows.

So far, many deadlock control policies based on siphon control have been proposed for Petri nets. The use of the $\max^*$-controllability condition to control a generalized Petri net is still an open problem requiring a further study. A sufficient and necessary siphon control condition for G-systems, S^4R, and S^*PR family remains open. Also, it is challenging to design an optimal supervisor for these generalized Petri nets based on siphons.

For AMS with unreliable resources, no much work is found on robust supervision design based on Petri nets. The thesis only seeks to offer simple or even naive solution for S^3PR. There is an obvious drawback in this study. The final supervisor for S^3PR is too complex in structure even though the algorithm is easily implemented in theory. Actually, robust liveness-enforcing supervisor design is much more difficult for complex AMS. Also, more effective methods with low computational overheads and simple structural complexity can be potentially studied in the future.

Bibliographie

[1] E. Badouel and P. Darondeau, Theory of Regions, *Lecture Notes in Computer Science*, vol.1491, pp.529-586, 1998. 23

[2] K. Barkaoui and I. B. Abdallah, Deadlock avoidance in FMS based on structural theory of Petri nets, *Proceedings of 1995 INRIA/IEEE Symposium on Emerging Technologies and Factory Automation*, vol.2, pp.499-510, 1995. 21

[3] K. Barkaoui and I. B. Abdallah, A deadlock prevention method for a class of FMS, *in Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, vol.5, pp.4119-4124, 1995. 21

[4] K. Barkaoui and J. F. Peyre, On livness and controlled siphons in Petri nets, *in Proceeding of the 17th International Conference on Application and Theory of Petri Nets, Lecture Notes in Computer Science*, vol.1091, pp.57-72, 1996. 24, 26, 27, 58, 69, 94

[5] K. Barkaoui, A. Chaoui, and B. Zouari, Supervisory control of discrete event systems based on structure theory of Petri nets, *IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation*, vol.4, pp.3750-3755, 1997. 89, 105

[6] K. Barkaoui, J. M. Couvreur, and K. Klai, On the equivalence between liveness and deadlock-freeness in Petri nets, *in Proceeding of the 26th International Conference on Application and Theory of Petri Nets, Lecture Notes in Computer Science*, vol.3536, pp.90-107, 2005. 22, 33

[7] F. Basile, P. Chiacchio, A. Giua, and C. Seatzu, Deadlock recovery of Petri net models

controlled using observers, *in Proceeding of the 8th IEEE International Conference on Emerging Technologies and Factory Automation*, pp.441-449, 2001. 25

[8] D. Y. Chao, Computation of elementary siphons in Petri nets for deadlock control, *Computer Journal*, vol.49, no.4, pp.470-479, 2006. 110, 118

[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. 25, 27, 58, 59, 66, 69, 77, 106

[10] D. Y. Chao, An incremental approach to extract minimal bad siphons, *Journal of Information Science and Engineering*, vol.23, no.1, pp.203-214, 2007. 111, 118, 119

[11] D. Y. Chao, Technical Note - MIP iteration-reductions for deadlock prevention of flexible manufacturing systems, *International Journal of Advanced Manufacturing Technology*, vol.41, no.3, pp.343-346, 2009. 25, 126

[12] D. Y. Chao, Conservative control policy for weakly dependent siphons in S^3PR based on elementary siphons. *IET Control Theory and Applications*, vol.4, no.7, pp.1298-1302, 2010. 111, 126

[13] Y. F. Chen and Z. W. Li, Design of a maximally permissive liveness-enforcing supervisor with compressed supervisory structure for flexible manufacturing systems, *Automatica*, vol.47, no.5, pp.1028-1034, 2011. 23, 24, 156

[14] Y. F. Chen, Z. W. Li, M. Khalgui, and O. Moshabi, Design of 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. 23

[15] Y. F. Chen, Z. W. Li, and M. C. Zhou, Behaviorally optimal and structurally simple liveness-enforcing supervisors of flexible manufacturing systems, *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol.42, no.3, pp.615-629, 2012. 23, 24

[16] F. Chu and X. Xie, Deadlock analysis of petri nets using siphons and mathematical programming, *IEEE Transactions on Robotics and Automation*, vol.13, pp.793-804, 1997 19, 20, 22, 24, 25, 27, 89, 100, 156

[17] E. G. Coffman, M. J. Elphick, and A. Shoshani, Systems deadlocks, *ACM Computing Surveys*, vol.3, no.2, pp.66-78, 1971. 20

[18] J. M. Colom and M. Silva, Improving the linearly based characterization of P/T nets, *Lecture Notes in Computer Science*, 483, 113-145, 1991. 51

[19] R. Cordone and L. Piroddi, Monitor optimization in Petri net control, *in Proceedings of the IEEE Conference on Automation Science and Engineering*, 24-27 Aug. 2011, Trieste, Italy, pp.413-418, 2011.

[20] R. Cordone and L. Piroddi, Parsimonious monitor control of Petri net models of flexible manufacturing systems, *IEEE Transactions on Systems, Man and Cybernetics : Systems*, vol.43, no.1, pp.215-221, 2013. 24

[21] J. Desel and J. Esparza, *Free Choice Petri Nets*, London : Cambridge University Press, 1995. 42

[22] J. Desel and W. Reisig, Place/transition Petri nets, Lectures on Petri nets I : Basic Models, *Lecture Notes in Computer Science*, vol.1491/1998, pp.122-173, 1998. 148

[23] J. Ezpeleta, J. M. Colom, and J. Martinez, A Petri net based deadlock prevention policy for flexible manufacturing systems, *IEEE Transactions on Robotics and Automation*, vol.11, no.2, 173-184, 1995. 16, 20, 21, 24, 25, 26, 43, 44, 48, 52, 83, 89, 100, 101, 123, 130, 132, 133, 135, 136, 156

[24] J. Ezpeleta, F. García-Vallés, and J. M. Colom, A class of well structured Petri nets for flexible manufacturing systems, *in Proceeding of the 19th International Conference on Applications and Theory of Petri Nets, Lecture Notes in Computer Science*, vol.1420, J. Desel and M. Silva (Eds.), pp.64–83, 1998. 19, 22, 83, 89

[25] J. Ezpeleta, F. Tricas, F. García-Vallés, J. M. Colom, A banker's solution for deadlock avoidance in FMS with flexible routing and multiresource states, *IEEE Transactions on Robotics and Automaton*, vol.18. no.4, pp.621–625, 2002. 89

[26] M. P. Fanti, G. Maione, and B. Turchiano, Digraph-theoretic approach for deadlock detec-

tion and recovery in flexible production systems, *Studies in Informatics and Control*, vol.5, no.4, pp.373–383, 1996. 22

[27] M. P. Fanti, G. Maione, and B. Turchiano, Deadlock detection and recovery in flexible production systems with multiple capacity resources, *in Proceedings of the 8th Mediterranean Electrotechnical Conference*, Bari, Italy, May 13–16, 1996, pp.237–241. 22

[28] M. P. Fanti, B. Maione, S. Mascolo, and B. Turchiano, Performance of deadlock avoidance algorithms in flexible manufacturing systems, *Journal of Manufacturing Systems*, vol.15, no.3, pp.164–178, 1996. 22

[29] M. P. Fanti, B. Maione, S. Mascolo, and B. Turchiano, Event-based feedback control for deadlock avoidance in flexible production systems, *IEEE Transactions on Robotics and Automation*, vol.13, no.3, pp.347–363, 1997. 22

[30] M. P. Fanti, B. Maione, and B. Turchiano, Event control for deadlock avoidance in production systems with multiple capacity resources, *Studies Informatics and Control*, vol.7, no.4, pp.343–364, 1998. 22

[31] M. P. Fanti, B. Maione, and B. Turchiano, Comparing digraph and Petri net ap- proaches to deadlock avoidance in FMS, *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol.30, no.5, pp.783–798, 2000. 22

[32] M. P. Fanti, G. Maione, and B. Turchiano, Distributed event-control for deadlock avoidance in automated manufacturing systems, *International Journal Production Research*, vol.39, no.9, pp.1993–2021, 2001. 22

[33] M. P. Fanti, Event-based controller to avoid deadlock and collisions in zone control AGVs, *International Journal of Production Research*, vol.40, no.6, pp.1453–1478, 2002. 22

[34] M. P. Fanti, G. Maione, and B. Turchiano, Design of supervisors to avoid deadlock in flexible assembly systems, *International Journal of Flexible Manufacturing Systems*, vol.14. no.2, pp.157–175, 2002. 22

[35] M. P. Fanti and M. C. Zhou Deadlock control methods in automated manufacturing systems,

IEEE Transactions on Systems, Man, and Cybernetics, Part A, vol.34, no.1, pp.5-22, 2004. 21, 22, 141

[36] A. Giua and C. Seatzu, Liveness enforcing supervisors for railway networks using ES^{2}PR Petri nets, *in Proceeding of the 6th International Work on Discrete Event System WODES'02*, pp.55-60, 2002. 25

[37] A. Giua and C. Seatzu, A systems theory view of Petri nets, Advances in Control Theory and Applications, *Lecture Notes in Control and Information Science*, C. Bonivento *et al*. (Eds.), vol.353, pp.99-127, 2007. 136

[38] 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. 23, 85

[39] F. S. Hsieh, Robustness of deadlock avoidance algorithms for sequential processes, *Automatica*, vol.39, pp.1695-1706, 2003. 30, 156

[40] F. S. Hsieh, Fault-tolerant deadlock avoidance algorithm for assembly processes, *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol.34, pp.65-79, 2004. 30, 156

[41] F. S. Hsieh, Robustness analysis of Petri nets for assembly/ disassembly processes with unreliable resources, *Automatica*, vol.42. no.7, pp.1159-1166, 2006. 30, 156

[42] F. S. Hsieh, Analysis of flexible assembly processes based on structural decomposition of Petri nets, *IEEE Transaction on System, Man and Cybernetics, Part A*, vol.37, no.5, pp.792-803, 2007. 30, 156

[43] F. S. Hsieh, Robustness analysis of holonic assembly/disassembly processes with Petri nets, *Automatica*, vol.44, pp.2538-2548, 2008. 30, 156

[44] F. S. Hsieh, Collaborative reconfiguration mechanism for holonic manufacturing systems, *Automatica*, vol.45, no.11, pp.2563-2569, 2009. 30, 156

[45] 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. 30, 156

[46] Y. S. Huang, M. D. Jeng, X. L. Xie, and S. L. Chung, A deadlock prevention policy for flexible manufacturing systems using siphons, *in Proceeding of IEEE International Conference on Robotics and Automation*, pp.541-546, 2001. 20, 21, 26, 83, 101, 156

[47] 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. 24, 25, 83, 89, 101, 130, 132

[48] 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*, vol.36, no.6, pp.2152-2160, 2006. 20, 25, 83, 89, 101

[49] Y. S. Huang and Y. L. Pan, An improved maximally permissive deadlock prevention policy based on the theory of regions and reduction approach, *IET Control Theory Application*, vol.5, no.9, pp.1069-1078, 2011. 120

[50] P. H. Starke, INA : Integrated Net Analyzer, http ://www2.informatik.hu - berlin.de/˜starke/ina.html, 2003. 38

[51] M. V. Iordache, J. O. Moody, and P. J. Antsaklis, Synthesis of deadlock prevention supervisors using Petri nets, *IEEE Transactions on Robotics and Automation*, vol.18, no.1, pp.59–68, 2002. 106

[52] M. D. Jeng, A Petri net synthesis theory for modeling flexible manufacturing systems, *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol.27, no.2, pp.169-183, 1997. 24, 64

[53] M. D. Jeng and X. L. Xie, Analysis of modularly composed nets by siphons, *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol.29, no.4, pp.399–406, 1999. 89

[54] M. D. Jeng, X. L. Xie, and M. Y. Peng, Process nets with resources for manufacturing modeling and their analysis, *IEEE Transactions on Robotics and Automation*, vol.18, no.6, pp.875–889, 2002. 89

[55] M. D. Jeng, X. L. Xie, and S. L. Chung, ERCN* merged nets for modeling degraded beha-

vior and parallel processes in semiconductor manufacturing systems, *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol.34, no.1, pp.102–112, 2004. 89

[56] M. D. Jeng and X. L. Xie, Deadlock detection and prevention of automated manufacturing systems using Petri nets and siphons, in Zhou, M. C. and Fanti, M. P. (Eds.) : *Deadlock Resolution in Computer-Integrated Systems (NY : Marcel-Dekker Inc. Press*, pp.233-281, 2005. 21, 156

[57] B. H. Krogh, R. Willson, and D. Pathak, Aotomated generation and evaluation of control programs for discrete manufacturing processes, *Computer Integrated Manufacturing*, pp.92-99, 1988. 38

[58] K. Lautenbach, Linear algebraic calculation of deadlocks and traps, In Voss, Genrich and Rozemberg (eds), *Concurrency and Nets*. Springer Verlag, Berlin, Germany, pp.315-336, 1987. 25

[59] M. A. Lawley, S. A. Reveliotis, and P. M. Ferreira, Design guidelines for deadlock handling strategies in flexible manufacturing systems, *International Journal of Flex- ible Manufacturing Systems*, vol.9, no.1, pp.5–30, 1997. 22

[60] M. A. Lawley, S. A. Reveliotis, and P. M. Ferreira, Flexible manufacturing system structural control and the neighborhood policy. Part 1. Correctness and scalability, *IIE Transactions*, vol.29, no.10, pp.877–887, 1997. 22

[61] M. A. Lawley, S. A. Reveliotis, and P. M. Ferreira, Flexible manufacturing system structural control and the Neighborhood Policy. Part 2. Generalization, optimization, and efficiency, *IIE Transactions*, vol.29, no.10, pp.889–899, 1997. 22

[62] M. A. Lawley, S. A. Reveliotis, P. M. Ferreira, A correct and scalable deadlock avoidance policy for flexible manufacturing systems, *IEEE Transactions on Robotics and Automation*, vol.14, no.5, pp.796–809, 1998. 22

[63] M. A. Lawley, S. A. Reveliotis, P. M. Ferreira, The application and evaluation of banker's algorithm for deadlock-free buffer space allocation in flexible manufactur- ing systems, *International Journal of Flexible Manufacturing Systems*, vol.10, no.1, pp.73–100, 1998. 22

[64] M. Lawley, Deadlock avoidance for production systems with flexible routing, *IEEE Transactions on Robotics and Automation*, vol.15, no.3, pp.497-509, 1999. 22

[65] M. A. Lawley, Integrating flexible routing and algebraic deadlock avoidance policies in automated manufacturing systems, *International Journal of Production Research*, vol.38, no.13, pp.2931–2950, 2000. 22

[66] M. A. Lawley and S. A. Reveliotis, Deadlock avoidance for sequential resource allocation systems : Hard and easy cases, *International Journal of Flexible Manufacturing Systems*, vol.13, no.4, pp.385–404, 2001. 22

[67] M. A. Lawley and W. Sulistyono, Robust supervisory control policies for manufacturing systems with unreliable resources, *IEEE Transactions on Robotics and Automation*, vol.18, no.3, pp.346–359, 2002. 22, 29

[68] 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. 64

[69] 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*, vol.34, no.1, pp.38-51, 2004. 20, 21, 24, 26, 28, 84, 101, 118, 130, 132, 156

[70] Z. W. Li and M. C. Zhou, Clarifications on the definitions of elementary siphons in Petri nets, *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol.36, no.6, pp.1227-1229, 2006. 26, 84

[71] Z. W. Li, J. Zhang, and M. Zhao, Liveness-enforcing supervisor design for a class of generalized Petri net models of flexible manufacturing systems, *IET Control Theory and Applications*, vol.1, no.4, pp.955-967, 2007. 20

[72] Z. W. Li, M. C. Zhou, and M. D. Jeng, A maximally permissive deadlock prevention policy for FMS based on Petri net siphon control and the theory of regions, *IEEE Transactions on Automation Science and Engineering*, vol.5, no.1, pp.182-188, 2008. 48

[73] Z. W. Li and M. C. Zhou, Control of elementary and dependent siphons in Petri nets and their application, *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol.38, no.1, pp.133-148, 2008. 126

[74] Z. W. Li, M. C. Zhou, and N. Q. Wu, A survey and comparison of Petri net-based deadlock prevention policy for flexible manufacturing systems, *IEEE Transactions on Systems, Man, and Cybernetics*, vol.38, no.2, pp.173-188, 2008. 102, 106

[75] Z. W. Li and M. C. Zhou, *Deadlock Resolution in Automated Manufacturing Systems : A Novel Petri Net Approach*, London : Springer Press, 2009. 28, 33, 45, 46, 47, 48, 53, 133, 134, 135

[76] Z. W. Li, S. Zhu, and M. C. Zhou, A divide-and-conquer strategy to deadlock prevention in flexible manufacturing systems, *IEEE Transaction on Systems, Man, and Cybernetics, Part C*, vol.39, no.1, pp.156-169, 2008. 48, 132, 133, 136, 137, 138, 139, 156

[77] Z. W. Li, H. M. Hanisch, and M. C. Zhou, Deadlock Prevention Based on Structure Reuse of Petri Net Supervisors for Flexible Manufacturing Systems, Techncial Reprot, No. SCATR201008001, Systems Control and Automation Group, School of Electro-Mechnical Engineering, Xidian University, Xi'an, China, 2010 (http ://sca.xidian.edu.cn/tech_reports.html). 89

[78] Lingo, Premier optimization modeling tools. http ://www.lingo.com/, 2013. 74, 77

[79] D. Liu, Z. W. Li , and M. C. Zhou, Liveness of an extended S^3PR, *Automatica*, vol.46, pp.1008-1018, 2010. 64

[80] G. Y. Liu, Z. W. Li, and C. F. Zhong, New controllability condition for siphons in a class of generalised Petri nets, *IET Control Theory and Applications*, vol.4, no.5, pp.854-864, 2010. 27, 58, 60, 61, 69, 106, 157

[81] G. Y. Liu and Z. W. Li, General mixed integer programming-based liveness test for system of sequential systems with shared resources nets, *IET Control Theory and Applications*, vol.4, no.12, pp.2867-2878, 2010. 15, 28, 39, 53, 54, 77, 78, 89

[82] G. Y. Liu, Z. W. Li, K. Barkaoui, and A. M. Al-Ahmari, Robustness of deadlock control for a class of Petri nets with unreliable resources, *Information Sciences*, vol.235, pp.259-279, 2013.

[83] G. Y. Liu and K. Barkaoui, Necessary and sufficient liveness condition of gs3pr petri nets, *International Journal of Systems Science*, DOI :10.1080/00207721.2013.827257269-278, 2014. 33

[84] J. L. Luo, W. M. Wu, H. Y. Su, and J. Chu, Supervisor synthesis for enforcing a class of generalized mutual exclusion constraints on Petri nets, *IEEE Transactions on Systems, Man, and Cybernetics*, vol.39, no.6, pp.1237–1246, 2009. 20

[85] T. Murata, Petri Nets : Properties, Analysis, and Applications, *in Proceeding of the IEEE*, vol.77, no.4, pp.541-580, 1989. 33, 40

[86] A. Nazeem, S. A. 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. 24

[87] A. Nazeem and S. A. Reveliotis, Designing compact and maximally permissive deadlock avoidance policies for complex resource allocation systems through classification theory : the nonlinear case, *IEEE Transactions on Automatic Control*, vol.57, no.7, pp.1670-1684, 2012. 24

[88] S. Park and J. Lim, Fault-tolerant robust supervisor for discrete event systems with model uncertainty and its application to a workcell, *IEEE Transactions on Robotics and Automation*, vol.15, pp.386-391, 1999. 29

[89] 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. 21, 24, 27, 53, 54, 89, 156

[90] L. Piroddi, R. Cordone, and I. Fumagalli, Selective siphon control for deadlock prevention in Petri nets, *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol.38, no.6, pp.1337-1348, 2008. 25, 28, 29, 106, 156

[91] L. Piroddi, R. Cordone, and I. Fumagalli, Combined siphon and marking generation for deadlock prevention in Petri nets, *IEEE Transactions on Systems, Man, and Cybernetics Part A*, vol.39, no.3, pp.650-661, 2009. 25, 29, 106

[92] P. J. Ramadge and W. M. Wonham, The control of discrete event systems, *Proceedings of the IEEE*, vol.77, no.1, pp.81-89, 1989. 22

[93] S. A. Reveliotis and P. M. Ferreira, Deadlock avoidance policies for automated manufacturing cells, *IEEE Transactions on Robotics and Automation*, vol.12, no.6, pp.845–857, 1996. 22

[94] S. A. Reveliotis, M. Lawley, and P. Ferreira, Polynomial complexity deadlock avoidance policies for sequential resource allocation systems, *IEEE Transactions on Automatic Control*, vol.42, pp.1344-1357, 1997. 21

[95] S. A. Reveliotis, Accommodating FMS operational contingencies through routing flexibility, *IEEE Transaction on Robotics Automation*, vol.15, pp.3-19, 1999. 29

[96] Y. Y. Shih and D. Y. Chao, Sequence of control in S^3PMR, *Computer Journal*, vol.53, no.10, pp.1691-1703, 2010. 110, 113, 118, 122, 123

[97] S. F. Chew and M. A. Lawley, Robust supervisory control for production systems with multiple resource failures, *IEEE Transaction on Automation Science and Engineering*, vol.3, no.3, pp.309-323, 2006. 29

[98] S. F. Chew, S. Y. Wang, and M. A. Lawley, Robust supervisory control for product routings with multiple unreliable resources, *IEEE Transaction on Automation Science and Engineering*, vol.6, no.1, pp.195-200, 2009. 29

[99] F. Tricas and J. Martinez, An extension of the liveness theory for concurrent sequential processes competing for shared resources, *in Proceeding of IEEE International Conference on Systems, Man, and Cybernetics*, pp.3035–3040, 1995. 89

[100] F. Tricas, F. García-Vallés, J. M. Colom, and J. Ezpeleta, An iterative method for deadlock prevention in FMS, *in Proceeding of 5th Workshop on Discrete Event Systems*, R. Boel and G.Stremersch (Eds.), pp.139-48, 2000. 27, 53, 54, 84, 89

[101] F. Tricas, Deadlock Analysis, Prevention and Avoidance in Sequential Resource Allocation Systems, Ph.D Dissertation, University of Zaragoza, Spain, 2003. 27

[102] F. Tricas and J. Ezpeleta, Some results on siphon computation for deadlock prevention in resource allocation systems modeled with Petri nets, *in Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation*, Lisbon, Portugal, 2003, pp.322–329. 27

[103] F. Tricas, F. Garcia-Vallès, J. M. Colom, and J. Ezpeleta, Using linear programming and the Petri net structure for deadlock prevention in sequential resource allocation systems, *XIII Jornadas de Concurrencia y Sistemas Distribuidos*, pp.65–77, 2005. 25, 27

[104] 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. 20, 23, 85, 104, 106, 156

[105] M. Uzam, The use of Petri net reduction approach for an optimal deadlock prevention policy for flexible manufacturing systems, *International Journal of Advanced Manufacturing Technology*, vol.23, no.3-4, pp.204-219, 2004. 23

[106] M. Uzam and M. C. Zhou, An improved iterative synthesis approach for liveness enforcing supervisors of flexible manufacturing systems, *International Journal Production Research*, vol.44, no.10, pp.1987-2030, 2006. 23, 91, 106

[107] M. Uzam, Z. W. Li, and M. C. Zhou, Identification and elimination of redundant control places in Petri net based liveness enforcing supervisors of FMS, *International Journal of Advanced Manufacturing Technology*, vol.35, no.1-2, pp.150-168, 2007. 23

[108] 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*, vol.37, no.3, pp.362-371, 2007. 23, 106

[109] N. Viswanadham, Y. Narahari, and T. L. Johnson, Deadlock prevention and deadlock avoidance in flexible manufacturing systems using Petri net models, *IEEE Transactions on Robotics and Automation*, vol.6, pp.713-723, 1990. 21

[110] S. G. Wang, C. Y. Wang, M. C. Zhou, and Z. W. Li, A method to compute strict minimal siphons in a class of Petri nets based on loop resource subsets, *IEEE Transaction System, Man, Cybernetics, Part A*, vol.42, no.1, pp.226-237, 2012. 111

[111] W. L. Winston and M. Venkataramanan, Introduction to Mathematical Programming, *Belmont CA : Duxbury Resource Center*, 2002. 25

[112] N. Wei and Z. W. Li, On the suboptimal liveness-enforcing supervisors based on Petri net structural analysis and the theory of regions, *International Journal of Advanced Manufacturing Technology*, vol.38, no.1-2, pp.195–204, 2008. 28

[113] 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*, vol.29, no.2, pp.192-204, 1999. 21, 22

[114] N. Q. Wu and M. C. Zhou, Avoiding deadlock and reducing starvation and blocking in automated manufacturing systems based on a Petri net model, *IEEE Transactions on Robotics and Automation*, vol.17 no.5, pp.658-669, 2001. 21

[115] N. Q. Wu and M. C. Zhou, Modeling and deadlock control of automated guided vehicle systems, *IEEE/ASME Transactions on Mechatronics*, vol.9, no.1, pp.50-57, 2004. 21, 156

[116] N. Q. Wu and M. C. Zhou, Modeling and deadlock avoidance of automated manufacturing systems with multiple automated guided vehicles, *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol.35, no.6, pp.1193-1202, 2005. 21, 156

[117] N. Q. Wu and M. C. Zhou, Deadlock resolution in automated manufacturing systems with robots, *IEEE Transactions on Automation Science and Engineering*, vol.4, no.3, pp.474-480, 2007. 21, 156

[118] 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.853-879, 1991. 22

[119] R. A. Wysk, N. S. Yang, and S. Joshi, Resolution of deadlocks in flexible manufacturing systems : avoidance and recovery approaches, *Journal of Manufacturing Systems*, vol.13, no.2, pp.128-138, 1994. 21, 22

[120] T. Kumaran, W. Chang, H. Cho, and A. Wysk, A structured approach to deadlock detection, avoidance and resolution in flexible manufacturing systems, *International Journal of Production Research*, vol.32, no.10, pp.2361–2379, 1994. 21

[121] X. L. Xie and M. D. Jeng, ERCN-merged nets and their analysis using siphons, *IEEE Transactions on Robotics and Automation*, vol.15, no.4, pp.692–703, 1999. 89

[122] K. Y. Xing, M. C. Zhou, H. X. Liu, 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*, vol.39, no.1, pp.188-199, 2009. 25, 84

[123] K. Y. Xing, M. C. Zhou, F. Wang, H. X. Liu, and F. Tian, Resource-transition circuits and siphons for deadlock control of automated manufacturing systems, *IEEE Transactions on System, Man, Cybernetics, Part A*, vol.41, no.1, pp.74-84, 2011. 84, 156

[124] M. Zhao, Z. W. Li, and H. S. Hu, Suboptimal liveness-enforcing supervisor design for a class of generalised Petri nets using partial siphon enumeration and mathematical programming, *International Journal of Systems Science*, vol.41, no.9, pp.1013-1026, 2010. 27, 28, 81

[125] C. F. Zhong and Z. W. Li, A deadlock prevention approach for flexible manufacturing systems without complete siphon enumeration of their petri net models, *Engineering with Computers*, vol.25, pp.269-278, 2009. 27, 28, 77

[126] 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. 27, 49, 50, 51, 54, 58, 59

[127] M. C. Zhou, F. DiCeasre, and A. A. Desrochers, A Top-down modular approach to synthesis of Petri net models for manufacturing systems, *in Proceeding of IEEE Robotics Automation Conference*, pp.534-539, 1989. 43, 44

[128] M. C. Zhou and F. DiCeasre, Adaptive design of Petri net controllers for error recovery in automated manufacturing systems, *IEEE Transactions on Systems, Man, and Cybernetics*, vol.19, no.5, pp.963-973, 1989. 43, 44

[129] M. C. Zhou and F. DiCeasre, Parallel and sequential mutual exclusions for Petri net modeling of manufacturing systems with shared resources, *IEEE Transactions on Robotics and Automation*, vol.7, no.4, pp.515-527, 1991. 43, 44

[130] M. C. Zhou, F. DiCesare, *Petri net synthesis for discrete event control of manufacturing systems.* London : Kluwer Academic Publishers, 1993. 21, 85

[131] M. C. Zhou, Deadlock avoidance methods for a distributed robotic system : Petri net modeling and analysis, *Journal of Robotic Systems*, vol.12, no.3, pp.177-187, 1995. 21

[132] M. C. Zhou and K. Venkatesh, *Modeling, Simulation, and Control of Flexible Manufacturing Systems : a Petri Net Approach*, Singapore : World Scientific, 1998. 33

[133] B. Zouari and K. Barkaoui, Parameterized supervisor synthesis for a modular class of discrete event systems, *in Proceeding of IEEE International Conference on Systems, Man, and Cybernetics*, Washington, DC, USA., 2003, pp.1874–1879. 89

[134] R. Zurawski and M. C. Zhou, Petri nets and industrial applications : a tutorial, *IEEE Transactions on in Industrial Electroniics*, vol.41, no.6, pp.567-583, 1994. 40

Glossaire

2^A	The power set of a set A.
C	A circuit.
C	A set of inequality constraints.
F	A set of flow relations in a Petri net.
f	A flow relations in a Petri net.
$H(r)$	The set of holders using resource r.
I	A place invariant.
$\|\|I\|\|$	The support of a place invariant I.
$\|\|I\|\|^+$	The positive support of a place invariant I.
$\|\|I\|\|^-$	The negative support of a place invariant I.
J	A transition invariant.
(l, b)	A general mutual exclusion constraint (GMEC).
(L, B)	A set of GMEC.
M	A marking.
$M(S)$	The sum of tokens in a place set S.
$M[t\rangle$	A transition t is enabled at M.
$max_{p^\bullet}$	$max\{W(p,t) \| t \in p^\bullet\}$
N	A Petri net with $N = \{P, T, F, W\}$.
$\mathbb{N}$	The set of non-negative integers, $\mathbb{N} = \{0, 1, 2, \ldots\}$.
$\mathbb{N}_m$	$\{1, 2, \ldots, m\}$.
$[N]$	The incidence matrix.
N_i	The ith Petri net.
$N_i \circ N_j$	Composition of nets N_i and N_j by shared places.
(N, M_0)	A Petri net system.
P	A set of places.
P^0	A set of idle places.
P_A	A set of operation (activity) places.
P_R	A set of resource places.
P_V	A set of additional places.
P_{U_r}	The set of unreliable places of r.
p	A place in a Petri net.
p^0	An idle place.

p_u	An unreliable place.
$p^{o\bullet}$	The set of transitions to which there are inhibitor arcs from place p.
$(p,t)^o$	An inhibitor arc.
$\overline{p}$	A complementary place.
$p^\bullet$	The postset of a place p.
${}^\bullet p$	The preset of a place p.
Post	The post-incidence matrix of a Petri net.
Pre	The pre-incidence matrix of a Petri net.
$R(N, M)$	The set of markings reachable from M.
$R(N, M_0)$	The set of markings reachable from M_0.
S	A siphon.
S_A	The set of activity places in S.
S_R	The set of resource places in S.
$[S]$	The complementary set of S.
T	A set of transitions in a Petri net.
t	A transition in a Petri net.
${}^\bullet t$	The preset of a transition t.
$t^\bullet$	The postset of a transition t.
${}^\bullet t^o$	The set of places from which there are inhibitor arcs to transition t.
W	A weight function.
$W(x,y)$	The weight of arc (x,y).
$W(f)$	The weight of arc f.
${}^\bullet x$	The preset of a node $x \in P \cup T$.
$x^\bullet$	The postset of a node $x \in P \cup T$.
X	A set.
$\|X\|$	The element count in a set X.
$\|\|X\|\|$	The cardinality in a set X.
${}^\bullet X$	The preset of a set $X \subseteq P \cup T$.
$X^\bullet$	The postset of a set $X \subseteq P \cup T$.
$\mathbb{Z}$	The set of integers.
Π	The set of strict minimal siphons.
Π_E	The set of elementary siphons.
Π_D	The set of dependant siphons.
Π_G	The set of plant siphon.
Π_H	The set of hybrid siphon.
Π_V	The set of monitor siphon.
Π_u	A set of uncontrolled siphons.
α_t	The times that t is fired from marking M.
σ	A firing sequence of a Petri net.
$\overrightarrow{\sigma}$	The Parikh vector of firing sequence σ.

Publications

Related Journal Papers :

[1] **Gaiyun Liu**, Zhiwu Li, Kamel Barkaoui, and Abdulrahman M. Al-Ahmari, Robustness of deadlock control for a class of Petri nets with unreliable resources, *Information Sciences*, vol.235, pp.259-279, 2013.

[2] **Gaiyun Liu** and Kamel Barkaoui, Necessary and sufficient liveness condition of GS^3PR Petri nets, *International Journal of Systems Science*, DOI : 10.1080/00207721.2013.827257(online), 2013.

[3] **Gaiyun Liu**, Daniel Yuh Chao, and Fang Yu, Control policy for a subclass of Petri nets without reachability analysis, *IET Control Theory and Applications*, vol.7, no.8, pp.1131-1141, 2013.

[4] Zhiwu Li, **Gaiyun Liu**, Hans-Michael Hanisch, and Mengchu Zhou, Deadlock prevention based on structure reuse of Petri net supervisors for flexible manufacturing systems, *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol.42, no.1, pp.178-191, 2012.

Related Conference Paper :

[5] **Gaiyun Liu**, Zhiwu Li, Abdulrahman M. Al-Ahmari, Liveness analysis of Petri nets using siphons and mathematical programming, *12th IFAC International Workshop on Discrete Event Systems in Paris*, 2014.

Résumé en Français

La révolution technologique dans notre monde réel nécessite de plus en plus de nouvelles techniques pour la synthèse et la vérification des systèmes complexes tels que les systèmes automatisés de production (AMS), les réseaux de communication, les systèmes embarqués temps-réel et les systèmes de contrôle du trafic. Un AMS est représenté sous sa forme logique par les systèmes à événements discrets (DES). Un AMS est mode de production avec une configuration commandée par ordinateur pour produire automatiquement des produits différents. Il existe trois systèmes principaux dans la plupart des AMS: (1) des machines de travail pour effectuer une série d'opérations, (2) un système de transport de matériel intégré avec un ordinateur pour contrôler le flux des matériaux, des outils et des informations dans tout le système, et (3) des stations de travail auxiliaires pour le chargement et le déchargement, le nettoyage, l'inspection…etc. Pour utiliser efficacement les précieuses ressources, elles doivent être partagées et soigneusement coordonnées entre les différents emplois concurrents. Le haut niveau de partage des ressources peut conduire à des conditions d'attente circulaires, la cause de blocage dans laquelle chaque ensemble de deux ou plusieurs tâches continue à attendre indéfiniment que les autres tâches de l'ensemble renoncent aux ressources qu'ils détiennent.

L’analyse et le contrôle de blocage jouent un rôle essentiel et critique dans la conception et le fonctionnement des AMS. Partiellement, les systèmes localement ou complètement paralysés par les blocages sont une situation hautement indésirable. Dans de nombreux cas, ils réduisent non seulement la productivité, mais aussi provoquent un coût économique fatal comme dans les systèmes de fabrication de semi-conducteurs et des résultats catastrophiques tels que les systèmes de manipulation d'une centrale nucléaire. Avec l'automatisation et la complexité croissante, la description, l'analyse, le contrôle et la résolution des blocages dans les AMS ont été des sujets d'un grand intérêt. Les réseaux de Petri sont bien appropriés pour décrire le comportement et les caractéristiques des AMS telles que la concurrence, les conflits, et la dépendance causale. Ils peuvent être utilisés pour révéler des propriétés comportementales telles que la vivacité et la borgnitude [16], [24]. Comparés aux automates à états finis qui sont

largement utilisés dans le cadre des DES, Les réseaux de Petri offrent une représentation compacte des DES car ils ne représentent pas explicitement l'espace d'état du système.

Au cours des dernières années, le contrôle d'exécution est devenu un domaine de recherche actif pour les AMS caractérisé par des processus hautement ordonnés avec des flux de données linéaires. Au cours des deux dernières décennies, de nombreux chercheurs ont considéré les réseaux de Petri comme une alternative aux automates. Les politiques de contrôle de blocage dans le cadre des réseaux de Petri peuvent être développées sur la base de l'analyse de l'espace d'état, l'analyse structurelle et la combinaison des deux anciennes méthodes. Traditionnellement, une politique de contrôle de blocage peut être évaluée par un certain nombre de critères de performance: complexité structurelle, la permissivité comportementale et la complexité de calcul.

L'analyse de l'espace d'état considère l'espace atteignable d'un modèle qui reflète le comportement possible de l'évolution du système. Les études dans [1], [104], [38], [105], [106], [107], [108] et [14] sont quelques-unes des œuvres les plus représentatives.

Dans [1], la théorie des régions est proposée. L'auteur vise à fournir une méthodologie formelle pour synthétiser un réseau de Petri à partit d'un système de transition. Plus tard, Uzam [104] propose une approche pour concevoir des superviseurs de réseaux de Petri optimaux en utilisant la théorie des régions. Peu de temps après [104], Ghaffari et al. présentent une explication facilement compréhensible de l'approche de conception d'un superviseur en termes de réseaux de Petri optimal basé sur la théorie des régions [38] et l'algèbre linéaire. Le travail dans [105] est une version améliorée de l'étude dans [104].

Si un superviseur optimal existe pour un modèle de réseaux de Petri, alors il peut être trouvé [104], [38]. Lorsqu'un superviseur optimal n'existe pas, le travail dans [104] et [38] n'offre pas une solution de contrôle de blocage. Dans ce cas, un problème intéressant est de trouver un superviseur de réseaux Petri qui soit le plus permissif possible. Le travail dans [14] présente une approche de prévention de blocage pour trouver un superviseur avec un maximum de permissivité pour un AMS si un tel superviseur existe. Sinon, il peut promouvoir le meilleur superviseur permissif posssible.

Chen et al. [13], [14], [15] développent une nouvelle méthode qui peut concrètement trouver un superviseur optimal en ajoutant des moniteurs. Cette méthode vise à bloquer le système non contrôlé de pénétrer dans la zone de blocage en empêchant l'atteignabilité de tous les premiers mauvais marquages (FBM). En outre, ils formulent une méthode pour s'assurer que

tous les marquages accessibles peuvent être atteints dans le système contrôlé et une technique pour réduire la charge de calcul en considérant uniquement un ensemble minimal de marquages accessibles et un ensemble minimal de FBM par une approche de recouvrement de vecteur.

Le travail dans [14] souffre du problème de la complexité structurelle car le nombre des places de contrôle calculé n'est pas minimal. Dans [13], les auteurs proposent une approche qui peut obtenir un superviseur permissif au maximum avec le nombre minimal de places de contrôle. C'est une approche non-itérative où toutes les places de contrôle peuvent être obtenues en résolvant un problème de nombres entiers en programmation linéaire (ILPP) (notée MCPP dans [13]). Bien que cette approche permette de surmonter les problèmes à la fois de la permissivité du comportement et de la complexité structurelle, elle souffre toujours du coût calcul élevé.

Le travail dans [15] emploie un petit (pas minimum) nombre de moniteurs mais plus efficace en surmontant le problème de la complexité des calculs dans [13]. Les auteurs réduisent le nombre de moniteurs en résolvant un ILPP à chaque itération, où un invariant de place est conçu pour une place de contrôle pour interdire autant que possible les FBM et permettre l'accessibilité de tous les marquages dans l'ensemble de couverture minimal des marquages accessibles. Ce résultat est obtenu en maximisant le nombre de FBM interdits par un invariant de place (PI) via la fonction objective de la ILPP. En supprimant les FBM interdits dans l'ensemble minimal couvert de FBM, ce processus est répété jusqu'à ce que tous les FBM soient interdits.

Ces contributions, si correctes et fiables, sont cependant loin d'être à la fine pointe de la littérature, étant donné qu'un meilleur modèle de ILP (plus ou moins de la même heuristique gloutonne) a été proposé dans [86]. Ce modèle dispose d'un certain nombre de contraintes qui sont linéaires par rapport au nombre d'états, par opposition à la quadratique requise par le modèle dans [13]. Si le modèle quadratique était plus serré que celui linéaire qui compenserait pour sa plus grande taille. Cependant, les résultats expérimentaux qui peuvent être obtenus implémentant à la fois les modèles sur un solveur de ILP et comparant leurs performances sur des instances de référence, le point vers une réponse négative.

De plus, les modèles ILP couramment adoptés dans la littérature ont des faiblesses théoriques, comme décrit dans [20], où il est montré comment plusieurs grandes instances de référence peuvent être résolues en quelques secondes pour garantir l'optimalité par un algorithme ad-hoc, alors qu'un solveur ILP nécessite des heures et souvent doit être résilié sans parvenir à

une garantie d'optimalité, pour ne pas mentionner la récente extension de cette théorie aux structures de contrôle et de surveillance plus complexes présentées dans [87]. Toutes les approches de thèses nécessitent l'analyse d'accessibilité et certains calculs pour calculer la vivacité critique et les marquages interdits.

La prévention de blocages basée sur le contrôle de siphons est une application typique des techniques d'analyse structurelle de réseaux de Petri. Les siphons, un objet de structure d'un réseau de Petri, sont largement utilisés pour analyser les problèmes de blocage dans les réseaux de Petri. Le contrôle de blocage en utilisant les siphons peut éviter le problème d'explosion d'états. Les chercheurs ont développé un grand nombre de politiques de contrôle de blocage basé sur le contrôle des siphons dont les œuvres représentatives sont données dans [23], [4], [16], [52], [89], [47], [69] , [103], [48], [9], [122], [90] et [91].

Dans [23], Ezpeleta et al. développent une méthode de conception de superviseurs de réseaux de Petri vivants à base de moniteur-AMS .Ce travail séminal est généralement considéré comme une contribution classique qui utilise des techniques d'analyse structurelle des réseaux de Petri pour éviter les blocages dans les AMS. Pour une classe typique de réseaux de Petri ordinaires, des systèmes de processus séquentiels simples avec des ressources (S^3PR), le travail dans [23] propose une politique de prévention de blocage en ajoutant une place de contrôle à chaque éventuel siphon strict minimal vide (SMS) pour s'auto-empêcher d'être vidé. L'importance de cette approche est qu'elle sépare avec succès un modèle de réseau d'usine et son superviseur. Cependant, il est coûteux en temps pour un modèle d'usine puisque le nombre de siphons dans un réseau accroit très rapidement et peut croître de façon exponentielle par rapport à sa taille [58]. En outre, l'approche [23] souffre des problèmes suivants: la permissivité comportementale, la complexité de calcul, et la complexité structurelle.

En raison de la complexité inhérente des réseaux de Petri, toute politique de prévention de blocage qui dépend d'une énumération complète de siphon est certainement exponentielle par rapport à la taille de son modèle d'usine. Dans [16], Chu et Xie utilisent une première programmation mixte en nombres entiers (MIP) pour détecter si un réseau de Petri structurellement limité n'a pas de blocages. Cette méthode évite l'énumération explicite de tous les siphons minimaux stricts et ouvre une nouvelle voie de recherche. Plus précisément, étant donné un réseau de Petri, un siphon non marqué maximal peut être obtenu par la solution de siphon traditionnelle suivante : D'abord, enlever toutes les places non marquées. Ensuite, retirer les transitions sans places d'entrée ainsi que leurs places de sortie. Répéter les

deux étapes jusqu'à ce qu'il n'y ait pas de places ni de transitions qui peuvent être enlevées. Une solution réalisable correspond à un siphon maximal non marqué lorsqu'il existe un siphon qui peut être vidé à un marquage qui est accessible à partir du marquage initial. Sinon, sa solution optimale est égale au nombre de toutes les places dans le réseau de Petri. Bien qu'un problème de MIP est NP-difficile en théorie [111], de nombreuses études numériques montrent que son efficacité de calcul est relativement insensible au marquage initial et est plus efficace que ceux qui dépendent de l'état complet ou de l'énumération du siphon. Le contrôle de blocage concerne habituellement les siphons minimaux. Huang et al. [47] proposent une politique de prévention de blocage en deux étapes itératives basées sur le travail dans [16]. A chaque itération, un siphon maximal non marqué est détecté par la résolution d'un problème de MIP. Si un tel siphon existe, alors un algorithme extrait un siphon strict minimal de celui maximal. Dans [7], [36] et [11], la technique de MIP est également utilisée. Leurs méthodes peuvent trouver directement un siphon minimal non marqué.

Un superviseur optimal basé sur une technique d'énumération complète de siphon souffre de la complexité structurelle élevée lorsque le nombre de siphons est grand. Ce problème a été connu depuis de nombreuses années. En utilisant pleinement la structure topologique d'un réseau de Petri, Les concepts de siphons élémentaires et dépendants dans un réseau de Petri sont proposés par Li et Zhou [69], [70]. Ils affirment que les siphons dans un réseau de Petri peuvent être divisés en ceux élémentaires et dépendants. Ces derniers peuvent être distingués ,en outre, par siphons fortement et faiblement dépendants à l'égard de ceux élémentaires. Il est montré que le nombre de siphons élémentaires dans un réseau est borné par la plus petite place et le nombre de transitions. Dans de nombreux cas, les moniteurs peuvent être ajoutés seulement aux siphons élémentaires. La contrôlabilité d'un siphon dépendant peut être assurée par la supervision du nombre initial de jetons dans les moniteurs qui sont ajoutés à ses siphons élémentaires. C'est-à-dire, un siphon dépendant peut implicitement être contrôlé en contrôlant ses siphons élémentaires corrélatifs. Ceci est illustré dans [69] par un exemple d'AMS. La contribution majeure de la théorie élémentaire est qu'elle abaisse particulièrement la complexité structurelle du superviseur. Notons que la méthode de [69] ne fait pas baisser la complexité de calcul ou améliorer la permissivité du comportement comparée à la politique en [23].

Dans un réseau de Petri ordinaire, un siphon est dit être contrôlé s'il ne peut être démarqué à tout marquage accessible [23], [46], [69]. Si un réseau de Petri est généralisé, cependant, la contrôlabilité d'un siphon est beaucoup plus complexe. En raison du poids des arcs,le fait

qu’un siphon ne soit pas vide n’est pas suffisant pour justifier l'absence de transitions mortes. L'existence d'un siphon strict minimal n’est plus nécessaire pour l'apparition de blocages. Dans l'ensemble, le concept de contrôlabilité concerne l'activation et le franchissement des transitions.

Comme classe typique de réseaux de Petri généralisés, un système de systèmes séquentiels avec des ressources partagées (S^4R) est proposé dans [4]. Il peut modéliser des systèmes d'allocation de ressources plus complexes avec de nombreux processus simultanés. Différents types de ressources multiples peuvent être demandées par différents procédés. Ainsi, un S^4R a une meilleure modélisation qu’un S^3PR qui est composé de machines à états et de ressources [23]. Par conséquent, la résolution d'un problème de contrôle de siphon pour S4R relève d’une importance dans la conception de superviseurs optimaux.

Cependant, le poids d'un arc dans un réseau de Petri généralisé peut être un nombre entier positif arbitraire de telle sorte qu'il est difficile de déterminer correctement la limite inférieure du nombre de jetons dans un siphon. Motivé par ce problème notoire, les chercheurs proposent un certain nombre de concepts impliquant la contrôlabilité de siphons dans un réseau de Petri généralisé, comme max-controlability [4], max′ - controlability [9], [9], [126], et max″- controlability [80]. La proposition de ces concepts vise à réduire le conservatisme d'une politique de prévention de blocage dont le développement repose sur le contrôle de siphons. En conséquence, un nombre suffisant mais les conditions de vivacité nécessaires ne sont pas développées. Cela nous motive à trouver un état de contrôlabilité plus général de siphons dans les réseaux de Petri généralisés.

Les systèmes généralisés de processus séquentiels simples avec des ressources (GS^3PR) sont une sous-classe de S^4R et une version généralisée d'un S^3PR. Il est facile de comprendre que les conditions de décision pour S^4R tiennent encore aux GS^3PR. La recherche sur la condition nécessaire et suffisante pour le contrôle de siphon dans GS^3PR sera un progrès important dans le contrôle de blocage des réseaux de Petri généralisés.

Les siphons sont bien connus pour être à égalité avec les blocages, ce qui est vrai dans les deux réseaux de Petri ordinaires et généralisés. La commande de blocage itérative est une stratégie classique en matière de prévention de blocage. Tricas utilise une approche d'itération pour éviter les blocages pour un AMS [100], [101], [102], [103]. A chaque étape d'itération, un siphon est calculé et commandé par un moniteur. Un tel procédé est poursuivi jusqu'à ce que tous les siphons soient contrôlés. Pour un S^4R, cette classe de politiques de prévention de

blocage itérative est généralement supposée converger à une certaine étape même si ce n'est pas une tâche facile de fournir une preuve formelle encore satisfaisante. Le travail a l'avantage d'éviter le problème d'explosion d'états. Cependant, une telle approche itérative, dans un cas général, conduit difficilement à un superviseur optimal en raison de la non maturité des techniques de contrôle du siphon pour les réseaux de Petri généralisés si les blocages sont éliminés à l'aide des concepts de max-controlled siphons [4] ou max'-controlled siphons [9].

Dans [124], basé sur les Deadly Marked Siphons (DMS) [89] dans well-marked S^4R, Zhao et al. modifient le test MIP dans [16] pour détecter les DMS pour les S^4R. Toutefois, un S^4R peut avoir des interblocages actifs même s'il est libre de blocages. Dans ce cas, les siphons causant l'interblocage ne peuvent pas être détectés par le MIP modifié et le réseau ne peut être contrôlé. En outre, les techniques à la fois dans [16] et [124] ne peuvent pas obtenir directement un siphon problématique minimal.

Dans [125], Zhong et al. proposent un modèle de MIP pour détecter un siphon minimal non-max-marqué [125]. Cependant, leur méthode ne peut pas détecter les siphons qui causent l'interblocage. En outre, il émet un SMS quand un réseau de Petri est vivant avec des siphons non-max-marqués créant une fausse impression que le réseau est non-vivant et qui donc a besoin d'une place de contrôle pour le contrôler.

Dans [81], les méthodes basées sur les MIP-existants sont améliorées dans la littérature en termes de max″-condition de contrôlabilité de siphons. Nous définissons les DMS (EDMS) étendus et développons ensuite un modèle de MIP plus général qui peut détecter les blocages et les interblocages causés par les siphons dans un S^4R. Nous concluons que le réseau est vivant si aucune solution n'est possible pour le modèle MIP. Cette programmation est plus puissante que les MIP dans [124] et [125], mais encore restrictive car elle émet un SMS quand un réseau de Petri est vivant avec des siphons non-max″-marked.

Récemment, plusieurs politiques de contrôle de blocage basées sur la combinaison de l'espace d'état et l'analyse structurelle ont été proposées. Le travail en [112] peut être considéré comme une amélioration de la théorie de régions. Il conçoit un superviseur pour un modèle avec un comportement permissif maximal en utilisant la théorie des régions. Ensuite, les SMS dans le système permissif maximal contrôlé sont calculés et divisés en ceux élémentaires et dépendants. Pour les empêcher d'être vidés, les expressions algébriques sur les marquages des moniteurs supplémentaires dans le superviseur et les places ressources dans le modèle de réseau sont dérivées, sous lesquelles le superviseur est vivant. Les expressions sont utilisées

pour calculer les marquages initiaux accessibles pour le superviseur sans changer sa structure lorsque le marquage de l'usine initial change. Une étude de cas montre que la méthode de calcul combinée est efficace comparée à celle existante dans laquelle la théorie des régions est utilisée seule et le comportement permissif du superviseur est presque optimal.

Dans [90], Piroddi et al. soulignent qu'il existe plusieurs inconvénients importants dans les méthodes de prévention de blocage qui sont basées sur les siphons élémentaires [69], [75]. Premièrement, les siphons élémentaires sont développés purement en utilisant la structure topologique d'un réseau, ne tenant pas compte de l'information de l'évolution dynamique du réseau. Deuxièmement, les politiques fondées sur des siphons élémentaires ne sont généralement pas permissives au maximum puisque les siphons contrôlés peuvent être contraints de garder plus d'un jeton. Troisièmement, l'ensemble des siphons élémentaires dans un réseau de Petri n’est pas unique. L'existence de différents ensembles de siphons élémentaires implique également que la solution de prévention de blocage n’est pas unique. Dernier point mais pas le moindre, les politiques fondées sur des siphons élémentaires peuvent être appliquées à certaines classes spéciales de réseaux de Petri seulement. Piroddi et al. croient qu'il est important d'intégrer les informations structurelles relatives aux siphons minimales strictes avec l'analyse de graphe d'accessibilité afin d'éviter les places de contrôle inutiles. Le travail dans [90] développe une politique de contrôle de siphon sélective dans laquelle les concepts de siphons dominés et dominants essentiels et critiques, les marquages dominants et dominés jouent un rôle important en résolvant les problèmes de recouvrement d'ensembles, les siphons dominants sont trouvés pour s’assurer que les siphons dominés sont contrôlés. Le superviseur résultant est très permissif. Le problème technique majeur dans [90] est sa complexité de calcul. A chaque itération, il faut calculer tous les siphons minimaux et tous les marquages qui dominent et résoudre un ensemble de problèmes couvrant , chacun des NP-difficile en théorie par rapport à la taille des réseaux. Plus tard, dans [91], Piroddi et al.améliorent la méthode en utilisant l'approche de détection de blocage basé sur les MIP de telle sorte que l’énumération complète minimale du siphon est évitée.

Toutes les études examinées dans la section précédente supposent que les ressources ne manquent pas. En fait, les échecs de ressources sont inévitables dans la plupart des AMS qui peuvent également provoquer le blocage d’un AMS. Ainsi, c’est une condition nécessaire pour élaborer une politique efficace et robuste de contrôle de blocage pour s’assurer que les blocages ne peuvent pas se produire même si certaines ressources dans un système sont en panne.

Il y a un manque de recherche dans les réseaux de Petri concernant les impacts des ressources fiables sur les AMS sous le contrôle de surveillance de blocages. En fait, les échecs de ressources sont un problème commun dans les systèmes du monde réel qui posent des défis en matière de contrôle prudentiel des systèmes à événements discrets dont les AMS. En cas de panne de ressources, les politiques de contrôle de blocage existantes sont toujours plus en vigueur et des blocages dans le système perturbé peuvent êtres causés. Par conséquent, une nouvelle analyse du système perturbé est généralement nécessaire. L'analyse de robustesse offre une autre façon de déterminer si l'opération d'un système perturbé ou une partie de celui-ci peut encore être maintenue en cas de défaillance de ressources. Au meilleur de notre connaissance, aucun travail sur le contrôle robuste des AMS basé sur les réseaux de Petri n'a été établi.

Reveliotis [95] considère un scénario où les pièces nécessitant une ressource défaillante peuvent être déroutées ou supprimées dans un système par une intervention humaine. Park et Lim [88] traitent des questions d'existence de superviseurs robustes. Lawley et al. [67], [97], [98] les superviseurs de conception pour les systèmes instables basé sur l'algorithme banquier et des contraintes du buffer central avec les propriétés suivantes: (1) le superviseur assure la production continue des types de pièces ne nécessitant pas de ressources échouées ; (2) le superviseur n'autorise que les états qui servent d'états initiaux en cas ou une panne de ressources supplémentaires se produit ; (3) le superviseur n'autorise que les états qui servent d'états initiaux si les ressources échouées sont réparées.

Hsieh développe une variété de méthodes pour déterminer la faisabilité de la production d'un jeu d'échecs de ressources modélisé comme l'extraction de jetons à partir d'un réseau de Petri [39], [40], [41], [42], [43], [44], [45]. Dans ces œuvres, des conditions de vivacité et l'analyse de la robustesse des réseaux sont basés sur les concepts de voies d'écoulement de jetons et les besoins en ressources minimales (MRR). Son travail apporte des conditions aux tolérances de pannes et propose une méthode de décomposition structurelle afin de tester la faisabilité des itinéraires de production. Cependant, toutes ces méthodes ne sont pas intuitives pour les modèles à réseaux de Petri. Dans cette thèse, nous essayons de faire respecter la vivacité et la robustesse par un superviseur en ajoutant des moniteurs et des sous-réseaux de récupération. Cela implique à la fois que le plan et son superviseur soient unifiés dans un formalisme des réseaux de Petri.

Une question intéressante est de savoir comment offrir aux politiques de contrôle de blocage existantes une propriété de robustesse souhaitable afin de faire face aux défaillances de

ressources. Plus précisément, la robustesse souhaitable est une propriété de système pour maintenir un système vivant contrôlé pour faire face à certaines ressources défaillantes. Dans cette thèse, nous nous concentrons sur une surveillance stricte de l'AMS. Nous espérons que le superviseur conçu pour l'AMS avec des ressources non fiables puisse avoir les propriétés suivantes: (1) il peut empêcher les blocages pour un modèle d'usine quand toutes les ressources fonctionnent normalement, (2) les blocages sont évités même si certaines ressources ne parviennent pas à travailler et sont enlevées pour réparation à tout moment et (3) les blocages disparaissent après que les ressources réparées soient retournées.

Cette thèse est destinée à fournir une solution au problème de blocage basé sur les réseaux de Petri et la conception d'un superviseur robuste pour les AMS. Les méthodes proposées s'appuient principalement sur l'analyse structurelle des réseaux de Petri. La thèse traite également des cas de ressources non fiables dans les AMS lors de la conception des contrôleurs de blocage.

En **introduction**, nous faisons d'abord un rappel sur les AMS et discutons de l'importance des problèmes de blocage et de leur résolution. Ensuite, nous examinons deux aspects différents des AMS, c'est à dire, la supervision optimale pour les AMS avec ou sans ressources peu fiables.

Le **chapitre 1** présente la définition formelle des réseaux de Petri et les concepts connexes, y compris les propriétés structurelles et comportementales telles que les invariants, les siphons, les pièges, la vivacité, et la borgnitude. Trois classes typiques de réseaux de Petri, c'est à dire, S^3PR, GS^3PR et S^4R sont introduites. Les relations entre elles sont également discutées. GS^3PR est une sous-classe de S^4R et une version généralisée de S^3PR. Ce chapitre est fondamental pour la compréhension des idées présentées dans les chapitres suivants.

Le **Chapitre 2** examine les concepts de siphons (max, max′, max″)-contrôlés et formule un nouveau concept appelé max*-controlled siphons pour les GS^3PR. Nous concluons qu'un GS^3PR est direct si et seulement si tous ses siphons stricts minimaux sont max* contrôlés. Puis des exemples sont donnés pour illustrer les siphons max, max′, max″, max* -contrôlés et leur différences par rapport à ceux qui existent déjà, le concept proposé est plus général. En outre, certains problèmes ouverts sont discutés.

Prenons le réseau de la figure. 1 comme exemple. Le réseau est vivant avec 654 marquages accessibles. *S* est non-max marqué à 8 marquages, non- max′-marqué à 4 marquages, et non max″-marqué à un marquage, comme indiqué dans le tableau 1, où le nombre de jetons dans

p_{11}, p_{12}, p_{13}, p_{14} est non représenté. Cependant, S est max*- marqué à tous les 654 marquages. En un mot, du siphon max-contrôlé au max*-contrôlé, les contraintes pour le contrôle du siphon deviennent de plus en plus faibles.

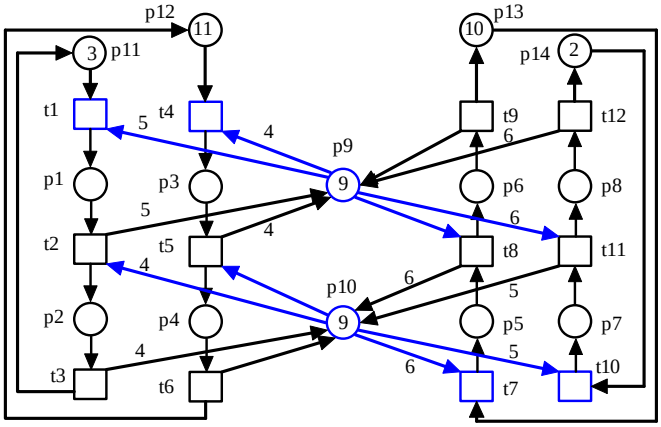


FIG. 1 – A live GS^{3}PR.

TAB. 1 – Controllability conditions of siphon $\{p_2, p_4, p_6, p_8, p_9, p_{10}\}$ in Fig. 1

markings of $p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}$	max	max′	max″	max•
1,0,1,0,1,0,0,0,0,3	no	no	no	yes
1,0,0,0,1,0,0,0,4,3	no	no	yes	yes
0,0,2,0,1,0,0,0,1,3	no	no	yes	yes
0,0,1,0,1,0,0,0,5,3	no	no	yes	yes
1,0,1,0,0,0,1,0,0,4	no	yes	yes	yes
0,0,1,0,0,0,1,0,5,4	no	yes	yes	yes
0,0,2,0,0,0,1,0,1,4	no	yes	yes	yes
1,0,0,0,0,0,1,0,4,4	no	yes	yes	yes

Basé sur la condition de max*-controllability ,nous pouvons concevoir des superviseurs ayant un comportement plus permissif pour les réseaux GS^{3}PR en théorie. Espérons que cette condition puisse être utilisée dans une stratégie appropriée qui permettra à un superviseur d'un système de réseaux de Petri d'être optimal. Ensuite, le système de fabrication correspondant est de plus en plus flexible. En ce sens, le nouveau concept proposé dans ce chapitre peut favoriser le développement du contrôle optimal ou sous-optimal de blocage. En d'autres termes, cette condition de vivacité nécessaire est suffisante pour les GS^{3}PR et sera considérée comme un progrès important dans le contrôle de blocage des réseaux de Petri généralisés. En raison des structures complexes de S^{4}R, trouver une condition de contrôle de siphon nécessaire et suffisante pour S^{4}R est un problème difficile qui reste ouvert et , qui, suivant notre propre opinion, a encore besoin de beaucoup d'efforts.

Basé sur la condition de max*-controllability des siphons, nous proposons un nouveau modèle de programmation en nombres entiers (IP) qui peut détecter des siphons non-max*- marqués qui causent directement des blocages ou inter-blocages. Nous concluons que s'il n'y a pas de solution possible à ce modèle, le réseau est vivant.

Prenez le réseau montré à la Fig. 1 comme exemple. La vivacité est vérifiée par la résolution de l'IP proposée. Nous utilisons Lingo [78] pour résoudre le problème de la propriété intellectuelle. Aucune solution réalisable ne peut être trouvée. Cela signifie que ce réseau est vivant. En fait, il y a un SMS unique dans ce réseau vivant: $S = \{p_2, p_4, p_6, p_8, p_9, p_{10}\}$ qui est toujours suffisamment marqué. Cependant, les MIP à la fois dans [125] et [81] peuvent trouver une solution réalisable pour ce réseau .en utilisant les MIP dans [125], le SMS S est non-max marqué au marquage $M = p_3 + p_5 + 5p_9 + 3p_{10} + 3p_{11} + 10p_{12} + 9p_{13} + 2p_{14}$. En utilisant les MIP dans [81], un siphon minimal S est non-max“-marqué au $M = p_1 + p_3 + p_5 + 3p_{10} + 2p_{11} + 10p_{12} + 9p_{13} + 2p_{14}$. Comme le montre le tableau 1, S sur la Fig. 1 est non-max-marqué à huit marquages et non-max“-marqué à un marquage. Par conséquent, une solution peut être trouvée. La politique de contrôle dans [125] ajoute un lieu de contrôle pour le SMS obtenu, ce qui n'est pas nécessaire dans la méthode proposée du chapitre en cours, comme indiqué par l'intermédiaire de cet exemple.

Le réseau représenté sur la Fig.2 a la même structure avec un réseau dans [81] et [9]. La Fig.3 est le graphe d'accessibilité du réseau de la Fig. 2. De toute évidence, nous pouvons voir que ce réseau est sans interblocages et déclare $M_9 = p_1 + p_2 + p_4 + p_5 + p_7 + p_{10}$ et $M_{10} = p_1 + p_2 + p_4 + p_5 + p_7 + p_8 + p_9$ sont des interblocages dans la Fig. 3.

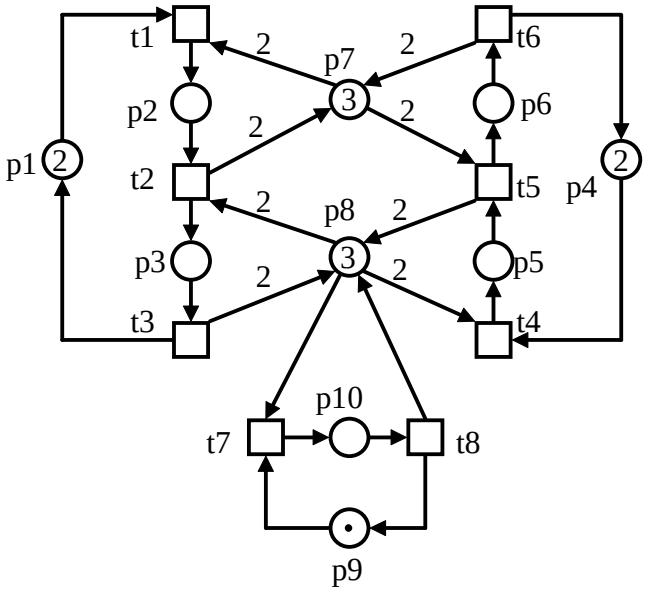


FIG. 2 – Un réseau dans [81].

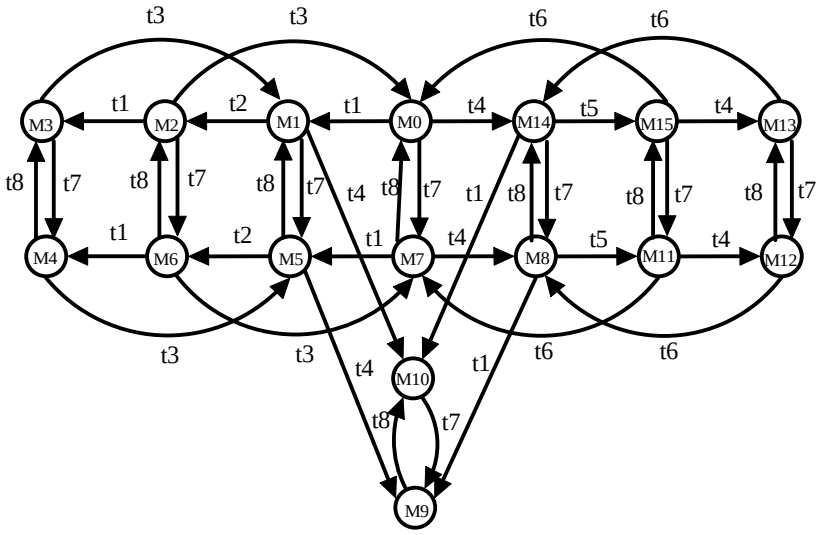


FIG. 3 – Le graphe d'accessibilité du réseau de la Fig. 2.

Le siphon minimal non-max*-marqué peut être détecté par le problème IP proposé. Le siphon minimal non-max*-marqué correspondant est $S=\{p_3, p_6, p_7, p_8, p_{10}\}$. En attendant, nous pouvons obtenir le mauvais marquage $M=p_1 + p_2 + p_4 + p_5 + p_7 + p_8 + p_9$. En contraste avec le graphe d'accessibilité dans la Fig.3, ce marquage est mauvais M_{10}. Pour Fig.2, le franchissement de t_7 à la fois aux marquages M_{10}, $M_9 = p_1 + p_2 + p_4 + p_5 + p_7 + p_{10}$ est alors obtenu. Une fois que le système évolue au M_9 ou M_{10} il ne peut jamais traiter d'autres processus. Ce sont des interblocages.

Pour cet exemple, on ne peut détecter un siphon mort marqué par la MIP proposée dans [124]. Aux mauvais marquages M_9 et M_{10}, le siphon $\{p_3, p_6, p_7, p_8, p_{10}\}$ n'est pas bloqué selon la définition des siphons bloqués. Plus précisément, t_7 et t_8 sont dans la présélection du siphon. La transition t_8 est activée par p_{10} au M_9 et t_7 est activée par p_8 au M_{10}. Celles-ci ne satisfont pas la condition $\forall t \in {}^{\bullet}S$, t est désactivé par certains $p \in S$.

En fait, par rapport à la technique de MIP dans [124], le projet IP est plus général dans les deux aspects: (i) un siphon minimal non-max*-marqué peut être obtenu directement si un réseau est non-vivant et (ii) la nouvelle adresse IP peut résoudre le problème si un réseau de Petri contient des inter-blocages causés par les siphons.

Depuis que l'approche est basée sur les siphons et la programmation mathématique, son efficacité de calcul est relativement insensible au marquage initial. Par rapport aux méthodes existantes, celle qui est proposée est plus puissante.

Le **Chapitre 3** présente une nouvelle méthode de conception des superviseurs de prévention de blocage basé sur les réseaux de Petri. Il ne garantit pas l'optimalité mais les résultats

empiriques montrent sa supériorité sur les autres approches fondées sur le contrôle de siphon. Compte tenu du modèle de réseau de Petri d'un AMS, on conçoit d'abord un système contrôlé de vivacité-d ’exécution optimale pour le modèle doté d'un marquage minimal initial en utilisant la théorie des régions. Ensuite, nous calculons tous les SMS dans le système contrôlé. Un tel siphon ne contient pas un piège. Pour chaque SMS, une inégalité algébrique respectant les marquages de moniteurs et des places ressources dans le système contrôlé, également appelée une contrainte vivacité, est établie en fonction de la notion de siphons max-contrôlés ou invariants contrôlés. Sa satisfaction implique l'absence de transitions mortes dans l’ensemble du siphon correspondant. Par conséquent, les marquages initiaux donnés qui répondent à toutes les contraintes d'inégalité de vivacité, tous les siphons peuvent être max-contrôlés et le système résultant est contrôlé vivant. Après avoir trouvé une structure asservie de système, on peut réaffecter les marquages initiaux en fonction des contraintes d'inégalité. Peu importe la taille des marquages initiaux et le nombre d'états, les contraintes de vivacité restent inchangées. Leur satisfaction garantit l'absence de siphons incontrôlés. Une fois la structure d'un système commandé de manière optimale est trouvée, un système quasi optimal peut être obtenu sans l'aide de la théorie des régions.

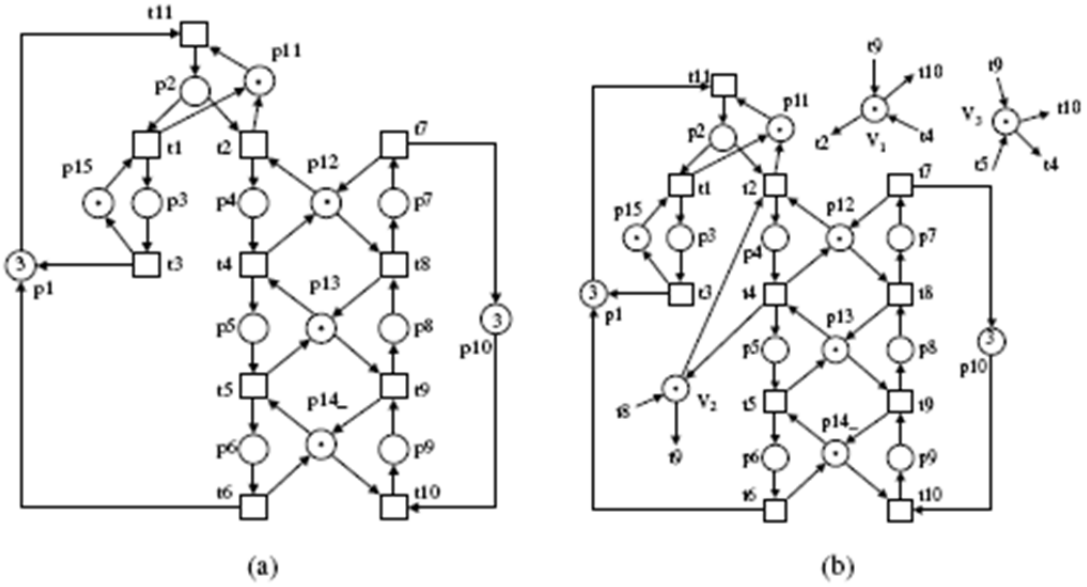


FIG. 4-(a) An M-net (N, M_0), (b) controlled system (N^{mc}, $M_0{}^{mc}$).

Prenons le modèle du réseau montré dans la Fig. 4(a) à titre d'exemple. C’est un M-réseau, où p_1 et p_{10} sont des places inactives, p_{11} - p_{15} sont des places ressources, et les autres sont des places actives. (N^{mc}, $M_0{}^{mc}$) représenté sur la Fig. 4(b) est le système contrôlé pour le réseau d’usine avec le marquage initial minimal. Sept siphons incontrôlés dans (N^{mc}, $M_0{}^{mc}$) $S_1 = \{p_6, p_8, p_{13}, p_{14}\}$, $S_2 = \{p_5, p_8, v_2, v_3\}$, $S_3 = \{p_5, p_7, p_{12}, p_{13}\}$, $S_4 = \{p_6, p_7, p_{12}, p_{14}, v_2, v_3\}$, $S_5 = \{p_6,$

p_8, p_{14}, v_2, v_3}, S_6 = {p_6, p_7, p_{12}, p_{13}, p_{14}} et S_7 = {p_5, p_7, p_{12}, v_2, v_3}. La forme de la matrice des contraintes de contrôlabilité est la suivante:

$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & -1 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} M_0^{mc}(v_1) \\ M_0^{mc}(v_2) \\ M_0^{mc}(v_3) \end{pmatrix} \leq \begin{pmatrix} 0 & 1 & 1 & -1 \\ 0 & 0 & 0 & -1 \\ 1 & 1 & 0 & -1 \\ 1 & 0 & 1 & -1 \\ 0 & 0 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ 1 & 0 & 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} M_0^{mc}(p_{12}) \\ M_0^{mc}(p_{13}) \\ M_0^{mc}(p_{14}) \\ 1 \end{pmatrix} \quad (1)$$

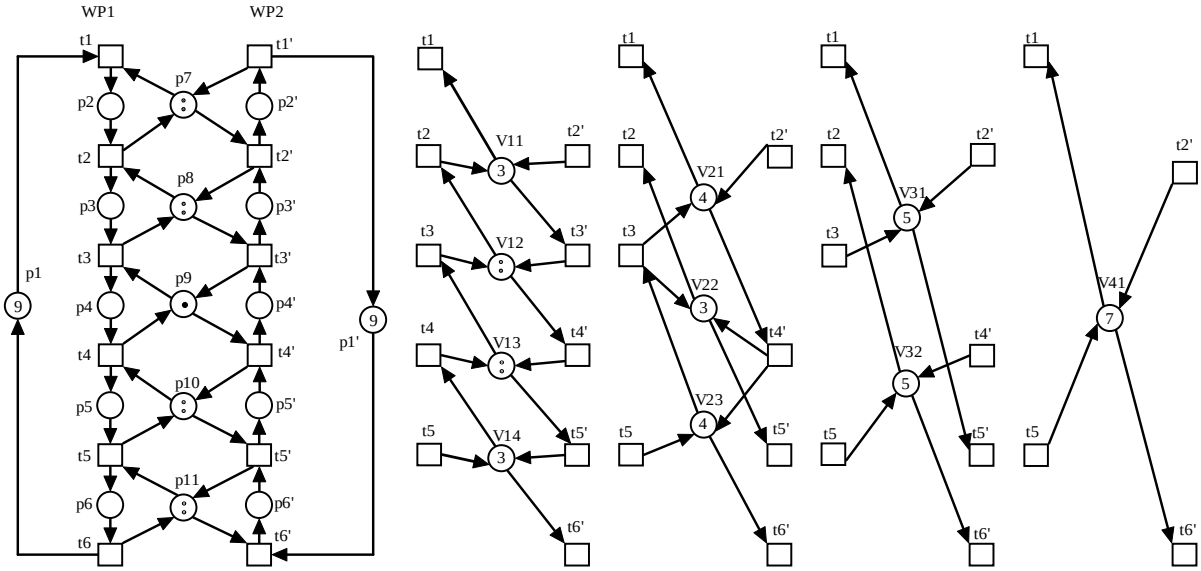
TAB. 2 – Behavioral permissiveness of the proposed deadlock prevention policy in Fig. 4(b)

$p_1, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}, p_{15}$	v_1, v_2, v_3	B_p	B_L	B_m	B_m/B_L
3, 3, 1, 1, 1, 1, 1	1, 1, 1	73	54	54	100%
4, 4, 2, 2, 2, 2, 2	3, 3, 2	1093	1047	941	89.88%
5, 5, 3, 3, 3, 3, 3	5, 5, 3	5767	5705	5151	90.29%
6, 6, 4, 4, 4, 4, 4	7, 7, 4	20324	20263	18517	91.38%
7, 7, 5, 5, 5, 5, 5	9, 9, 5	57450	57390	52995	92.25%
8, 8, 6, 6, 6, 6, 6	11, 11, 6	140703	140643	131000	93.14%
9, 9, 7, 7, 7, 7, 7	13, 13, 7	310783	310723	291363	93.77%
10, 10, 8, 8, 8, 8, 8	15, 15, 8	634173	634113	597853	94.28%
11, 11, 9, 9, 9, 9, 9	17, 17, 9	1214679	1214619	1150189	94.70%
12, 12, 10, 10, 10, 10, 10	19, 19, 10	2208445	2208385	2098887	95.04%

Le tableau 2 montre le comportement des systèmes contrôlés permissifs à différents marquages initiaux où les marquages initiaux des moniteurs sont décidés par l'Eq. (1). Dans ce tableau, B_p est le nombre d'états accessibles de (N, M_0), B_L représente le nombre d'états d'un système contrôlé optimal pour (N, M_0) a, B_m indique le nombre d'états du système contrôlé (N^{mc}, $M_0{}^{mc}$), et B_m/B_L implique le degré d'optimalité.

Le **chapitre 4** propose une politique permissive maximale de commande pour une sous-classe de S^3PR (appelé β-filets) basée sur la théorie du modèle de distribution de jetons des siphons. Nous montrons d'abord, que par adjonction d'un moniteur pour chaque siphon critique, certains états vivants peuvent se perdre quand le moniteur contrôle l'ensemble complémentaire d'un siphon critique et certaines places ne peuvent être marquées. En ne contrôlant que l'ensemble des places de fonctionnement marquées, les états plus vivants peuvent être atteints. Cependant, ce qui induit à des siphons vides. Le modèle de jetons correspondant peut être déduit. En ajoutant les moniteurs à tous ces siphons éventuellement vides, le réseau controlé devient vivant et permissif au maximum. Il n' est pas nécessaire de construire un graphe d'accessibilité et d'énumérer tous les siphons minimaux.

Pour le réseau dans la Fig. 5 (a), il existe quatre circuits de base à partir desquels, on peut synthétiser S_1, S_2, S_3, S_4 et 1- quatre écrans de V_{11}, V_{12}, V_{13}, V_{14} et quatre siphons de base sont ajoutés en conséquence, comme illustré dans la Fig. 5(b). Pour chaque siphon 2-dépendant, il n'y a pas place bornée et aucune opération d'échange n'est disponible. Les Moniteurs V pour ces deux siphons-dépendants sont ajoutés (Fig. 5(c)) de sorte que [V] = [S]. Fig. 5(d) et (e) montrent les 3 et 4 moniteurs avec [V] = [S]. Le modèle contrôlé final atteint 2566 États.



(a) A Petri net (N', M'_0) (b) 1-monitor (c) 2-monitor (d) 3-monitor (e) 4-monitor

FIG. 5–An example of the control policy based on UP but with no unmarked places in $H(V)$.

Pour atteindre plusieurs états, examiner les opérations d'échange pour les 3-siphon dépendant S'_1 et S'_2. $S_{R'1}$ ={p_8, p_9} a seulement une place bornée p_9 avec une opération d'échange déplaçant un jeton de p'_3 p'_4 de sorte que le modèle de jeton passe de $M'_\Psi = 2p_2 + 2\ (p_3 \oplus p'_3) + 2p'_5$ à $M'_{\Psi'} = 2p_2 + p_3 + p'_4 + 2p'_5$. Un moniteur V est ajouté pour chacun des Ψ et Ψ′ avec $M_0\ (V) = M_0(S) - \theta - 1 = 7-2 = 5$. $S_{R''2}$ = {p_9, p_{10}} a une seule place bornée avec une opération d'échange consistant à déplacer un jeton à partir de p_5 p_4 de sorte que le modèle de jeton passe de $M'_{\Psi^*} = 2p_3 + 2\ (p_5 \oplus p'_5) + 2p'_6$ à $M'_{\Psi\wedge} = 2p_3 + p_4 + p'_5 + 2p'_6$. Ajouter un moniteur V à chacun des Ψ* et Ψ^ avec $M_0\ (V) = M_0(S) - \theta - 1 = 7-2=5$. Ces deux opérations d'échange conduisent à 2606 > 2566 états vivants.

Il y a seulement un 4-siphon dépendant S. S_R = {p_8, p_9, p_{10}} qui a une seule place bornée p_9 mais avec deux opérations d'échange possibles. Tout d'abord une déplace un jeton de p'_3 à p'_4 de sorte que le modèle de jeton passe de $M'_\Psi = 2p_2 + 2\ (p_3 \oplus p'_3) + 2\ (p_5 \oplus p'_5) + 2p'_6$ à $M'_{\Psi'} = 2p_2 + p_3 + p'_4 + 2\ (p_5 \oplus p'_5) + 2p'_6$. Ajoute un moniteur pour chacun des V et Ψ Ψ′ avec

$M_0(V) = M_0(S) - \theta - 1 = 9-2 = 7$. On notera que lorsque $M_0(p_9) = 2$, toutes les adresses internes sont non singulières et tous les siphons critiques sont ceux composés. Le marquage pour chaque S non-marqué est un UP.

TAB. 3 – Controlled model for the net in Fig. 5.

S	$V(M_0)$	$V_S^\bullet$	${}^\bullet V_S$	$[V_S]$
S_1	$V_{11}(3)$	t_1, t'_3	t_2, t'_2	p_2, p'_3
S_2	$V_{12}(2)$	t_2, t'_4	t_3, t'_3	p_3, p'_4
S_3	$V_{13}(2)$	t_3, t'_5	t_4, t'_4	p_4, p'_5
S_4	$V_{14}(3)$	t_4, t'_6	t_5, t'_5	p_5, p'_6
S_5	$V_{21}(4)$	t_1, t'_4	t_3, t'_2	p_2, p_3, p'_3, p'_4
S_6	$V_{22}(3)$	t_2, t'_5	t_3, t'_4	p_3, p'_5
S_7	$V_{23}(4)$	t_3, t'_6	t_5, t'_4	p_4, p_5, p'_5, p'_6
S_8^1	$V_{31}^1(5)$	t_1, t'_5	t_3, t'_3	p_2, p_3, p'_4, p'_5
S_8^2	$V_{31}^2(5)$	t_1, t'_3, t'_5	t_3, t'_2, t'_4	p_2, p_3, p'_3, p'_5
S_9^1	$V_{32}^1(5)$	t_2, t'_6	t_4, t'_4	p_3, p_4, p'_5, p'_6
S_9^2	$V_{32}^2(5)$	t_2, t_4, t'_6	t_3, t_5, t'_4	p_3, p_5, p'_5, p'_6
S_{10}^1	$V_{41}^1(7)$	t_1, t_4, t'_6	t_3, t_5, t'_3	$p_2, p_3, p_5, p'_4, p'_5, p'_6$
S_{10}^2	$V_{41}^2(7)$	t_1, t_4, t'_3, t'_6	t_3, t_5, t'_2, t'_4	$p_2, p_3, p_5, p'_3, p'_5, p'_6$
S_{10}^3	$V_{41}^3(7)$	t_1, t'_3, t'_6	t_4, t'_2, t'_4	$p_2, p_3, p_4, p'_3, p'_5, p'_6$

V_{ij} indicates V is the j-th i-dependent critical siphon. V^k $(k > 1)$ indicates V is a EUP

Le second déplace un jeton de p_5 p_4 (cela ne change pas le nombre total de jetons dans $[S]$) de telle sorte que le modèle de jetons change de $M'_\Psi = 2p_2 + 2(p_3 \oplus p'_3) + 2(p_5 \oplus p'_5) + 2p'_6$ à $M'_{\Psi\wedge} = 2p_2 + 2(p_3 \oplus p'_3) + p_4 + p'_5 + 2p'_6$. Ajouter un moniteur V pour Ψ avec $M_0(V) = M_0(S) - \theta - 1 = 9-2 = 7$. Il y a trois moniteurs supplémentaires ci-dessus. Le modèle résultant comme indiqué dans le tableau 3 est permissif et atteint au maximum 2628 États.

Le **Chapitre 5** comble l'écart entre une stratégie de contrôle de blocage « diviser pour régner » et son application aux systèmes du monde réel avec des ressources peu fiables. Pour S^3PR, les sous-réseaux de récupération et les moniteurs sont conçus pour les ressources non fiables et les siphons minimaux stricts qui peuvent être vidés, respectivement. Les arcs ordinaires et inhibiteurs sont utilisés pour connecter des moniteurs avec des sous-réseaux de récupération en cas de nécessité. Puis la ré-analyse du réseau de Petri originale est évitée et un superviseur optimal robuste est dérivé. Les superviseurs conçus pour S^3PR par la méthode proposée ont les propriétés suivantes: (1) ils peuvent empêcher les blocages pour les modèles d'usine quand toutes les ressources fonctionnent normalement; (2) les blocages sont évités, même si certaines ressources ne parviennent pas à travailler et sont à réparer à tout moment et

(3) les états d'attente pour la réparation disparaissent après que les ressources réparées soient retournées.

Une cellule de fabrication automatisée représentée dans la Fig. 6 (a) présente un outil *M* et un type de robots *R* avec deux robots. Chacun des robots peut se déplacer et l'outil peut traiter une partie à la fois. Les pièces entrent dans la cellule par l'intermédiaire du le buffer I, et quittent la cellule à travers le buffer O. l'outil exerce une activité sur les pièces brutes et les robots traitent les mouvements des pièces. La cellule peut être modélisée avec les réseaux de Petri comme le montre la Fig. 6 (b). Le réseau est un S^3PR, où $P^0 = \{p_1\}$, $P_R = \{p_5, p_6\}$, et les autres sont des places d'activité.

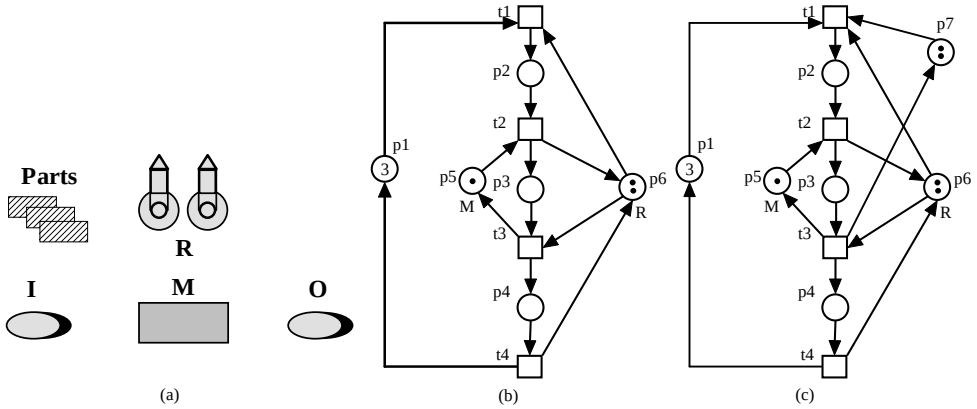


FIG.6 –(a) An auyomated manufacturing cell, (b) a Petri net model, (c) a liveness-enforcing supervisor.

Dans la Fig. 6(b), $S = \{p_4, p_5, p_6\}$ est un SMS unique. Le siphon est vide au $2p_2 + p_3$ et corrélativement le système est dans un état de blocage. Physiquement, cela signifie que chaque robot est maintenu et la machine traite une partie en même temps. Toutes les ressources sont occupées et sont en attente circulaire. Nous devons éviter ce cas de telle sorte que le système puisse continuer à fonctionner. En utilisant les méthodes traditionnelles dans [23], [47] et [69] qui ajoutent des moniteurs pour siphons, nous pouvons trouver un superviseur optimal pour les S^3PR, comme le montre la Fig. 6 (c).

Les politiques de contrôle de blocage dans [23], [47] et [69] sont développées en supposant que les ressources dans un système sont fiables. Cependant, les AMS du monde réel souffrent souvent de défaillances de ressources non fiables. Dans la Fig. 6(a), nous supposons qu'un robot tombe en panne au marquage initial. Ensuite, le robot qui a un mauvais fonctionnement doit être enlevé et récupéré. Un jeton est en conséquence retiré de p_6 la Fig. 6(c), puis le réseau montré sur la Fig. 7 est dérivé.

Dans la Fig. 7 (b), marquant $p_1 + p_2 + p_3$ est un état de blocage. Moniteur p_7 ne peut pas contrôler le système dans ce cas. En d'autres termes, le superviseur du système n'est pas robuste dans le sens où un robot cassé conduit le système d'origine contrôlé à un état de blocage. Il est nécessaire de concevoir un superviseur de vivacité-d'exécution robuste pour le réseau de Petri, qui permet de contrôler le système si un robot tombe en panne ou non.

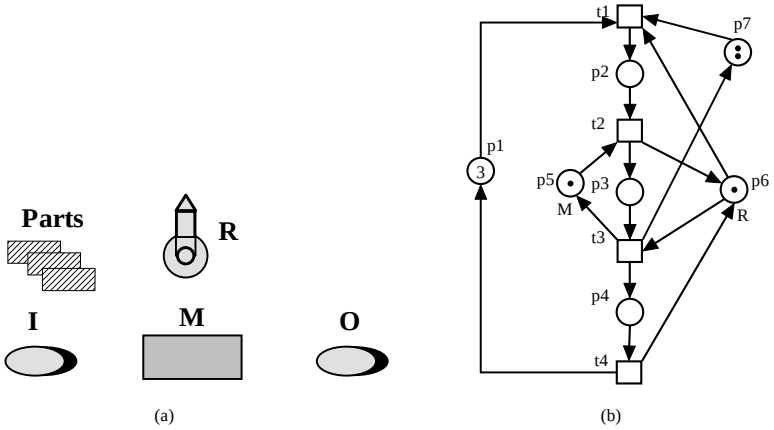


FIG.7 – (a) One robot is removed, (b) one token is correspondingly removed from p_6.

Comme le montre la Fig. 8, nous retirerons un jeton du moniteur p_7 quand un jeton est retiré de p_6, t_7 est tirée. Cela signifie que le robot avec un dysfonctionnement est envoyé pour être réparé. Dans ce cas, Le moniteur p_7 avec un jeton rend le système contrôlable. En fait, pour le type de ressource R, les robots peuvent se décomposer dans p_2, p_4, ou p_6. Une fois qu'une ressource se décompose à une certaine place, elle doit être retirée,ce qui ne dépend pas du fait que son moniteur marqué soit lié ou non. Basé sur ce fait, un superviseur de vivacité-d'exécution robuste est conçu comme dans Fig. 8.

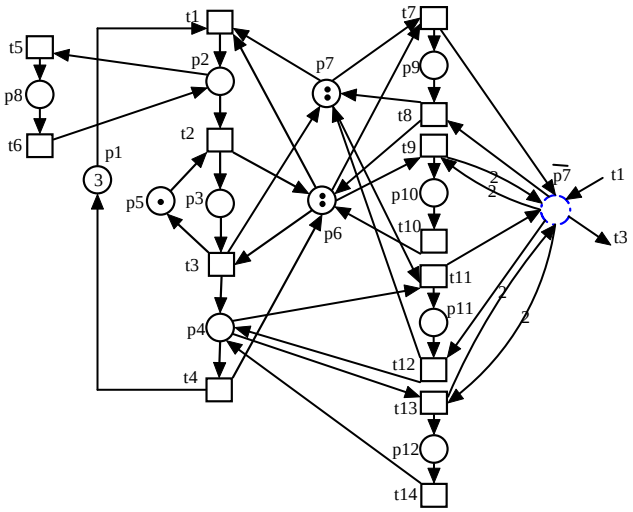


FIG. 8 – Un superviseur robuste d'un système.

En effet, les échecs de ressources sont de hasard. Plus précisément, les ressources qui échoueront, les états auquels les ressources vont échouer et le nombre de ressources est brisé avant que des erreurs se produisent. L'objectif de ce chapitre est de concevoir un superviseur robuste qui peut gérer les erreurs autant que possible sans une nouvelle analyse du système d'origine. Cependant, basé sur l'algorithme proposé, le superviseur de vivacité-d'éxecution robuste pour un S^3PR a une structure complexe. Nous montrons que la structure de ce superviseur peut être réduite par l'introduction d'arcs inhibiteurs.

Le classe des réseaux de Petri étudiée dans ce chapitre est celle des réseaux délimités. Un réseau de Petri délimité avec des arcs inhibiteurs peut être transformé en un réseau de Petri classique [22]. Compte tenu d'une k-place-bornée p avec ${}^{\bullet}p = T^* = \{t_1, \cdots, t_m\}$, $p^{\bullet} = T^{*\prime} = \{t'_1, \cdots, t'_n\}$, et $p^{o\bullet} = \{t\}$, le comportement de l'arc inhibiteur $(p, t)^o$ peut être remplacé par un équivalent en ajoutant une place complémentaire p avec ${}^{\bullet}\overline{p} = T^{*\prime} = \{t'_1, \cdots, t'_n\}$, $\overline{p}^{\bullet} = T^* = \{t_1, \cdots, t_m\}$ pour p et deux arcs normaux de and vers $\overline{p}$, pondérés par k. Pour chaque marquage accessible, p est vide si et seulement si $\overline{p}$ porte k jetons. Puis l'arc inhibiteur est remplacé. La Fig. 9 illustre cette transformation équivalente. Cette propriété d'arcs inhibiteurs de réseaux de Petri délimités est importante pour nous afin de simplifier la structure de supervision dérivée de l'algorithme original proposé.

Ensuite, on a $(N^{rc^*}{}_V, M^{rc^*}{}_{0V})$ pour le S^3PR d'origine, comme le montre la Fig. 10. En comparant la Fig. 10 avec 8, la place complémentaire $\overline{p}$ et les arcs connexes sont remplacés par un arc inhibiteur. La structure du superviseur est simplifiée.

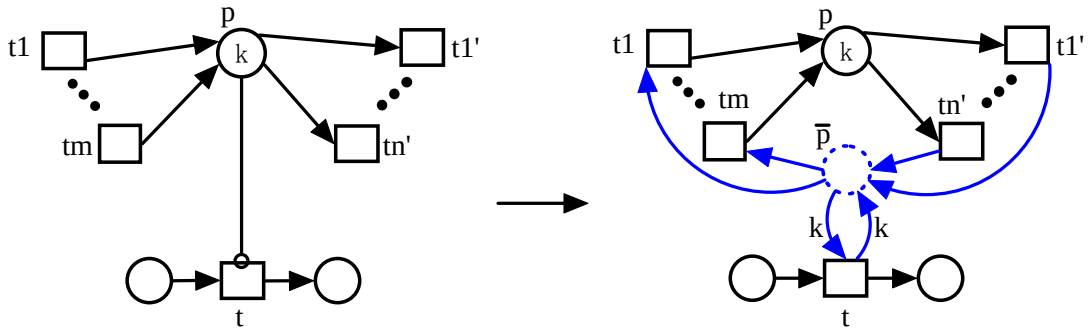


FIG. 9 – Transformation.

Comparés aux politiques traditionnelles de commande de blocage, les avantages de la méthode proposée sont que la nouvelle analyse du réseau peut être évitée. Dans une large mesure, la robustesse du superviseur est améliorée. Les résultats indiquent que le contrôleur est qualifié avec la robustesse et la vivacité. Cependant, il existe un inconvénient évident dans

cette étude. Le superviseur final pour S^3PR est une structure trop complexe, même si l'algorithme est facilement mis en œuvre pour une S^3PR en théorie.

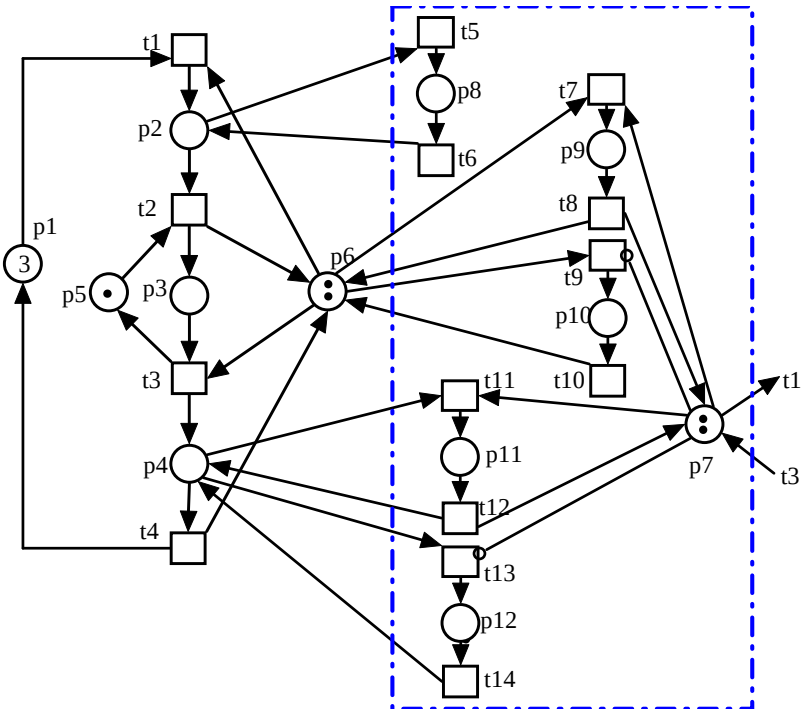


FIG. 10 – Superviseur avec arcs inhibiteurs.

Enfin, nous résumons cette thèse en soulignant les contributions majeures. Cette thèse examine certaines questions de contrôle de blocage importants dans les AMS basé sur réseaux de Petri, l'analyse structurelle et la conception robuste du superviseur. Trois défis sont abordés. Le premier est la dérivation de la perte de contrôlabilité des siphons. Le second est la conception de superviseurs avec un comportement semi-optimal/optimal et permissif et une faible complexité de calcul. Le troisième est la conception d'un superviseur robuste optimal pour les AMS avec des ressources peu fiables. Le premier défi est relevé en proposant le concept de max*-controllability et les techniques de test de l'IP. Le deuxième problème est abordé en combinant la théorie des régions et celle de l'analyse structurelle. En outre, une politique de contrôle au maximum permissive pour une sous-classe de réseaux de Petri basé sur la théorie du modèle de distribution de jetons de siphons est proposée. Il n'y a pas besoin de construire un graphe d'accessibilité et d'énumérer tous les siphons minimaux. La troisième question est surmontée par l'introduction de sous-réseaux de Petri. Un superviseur optimal robuste est conçu de telle sorte qu'un bon compromis entre les politiques de contrôle de réseaux Petri existantes et leur application aux systèmes du monde réel avec des ressources non fiables peut être fait. Les limitations et questions de recherche futures sont citées. Plusieurs problèmes non résolus et intéressants sont illustrés.

Gaiyun LIU

Supervisor Synthesis for Automated Manufacturing Systems Based on Structure Theory of Petri Nets

Résumé :
Le contrôle de systèmes industriels à cause de l'automatisation et la réduction de nombre des opérateurs devient un enjeu crucial. Les systèmes de production automatisés (AMS) sont d'autant plus touchés car une défaillance du programme de contrôle peut réduire considérablement la productivié voire entraôner l'arrêt du système de production. Pour certains de ces systèmes où le partage des ressources est pondérant, la notion de blocage partiel ou global est fréquente et la validation avant implantation est préférable pour réduire les risques.
En raison de la capacité des réseaux de Petri à décrire aisément l'exécution concurrente des processus et le partage des ressources, de nombreuses méthodes de vérification d'absence de blocage et de synthèse de contrôleurs basées sur la théorie structurelle ou le graphe d'accessibilité des réseaux de Petri ont été proposées au cours des deux dernières décennies. Les méthodes fondées sur l'espace d'état aboutissent généralement à un contrôle maximal permissif mais souffrent de l'explosion combinatoire de l'espace d'états. En revanche, les méthodes de synthèse de contrôleurs fondées sur l'analyse structurelle évitent le problème de l'explosion de l'espace d'état mais aboutissent à des superviseurs pouvant restreindre considérablement les comportements admissibles du système. De plus si la théorie structurelle de contrôle de siphons pour la synthèse des superviseurs est mature dans le cas des réseaux de Petri ordinaires, elle est en développement pour les réseaux de Petri généralises. Par ailleurs, la plupart des travaux existants partent du principe que les ressources sont constamment disponibles. Or l'indisponibilité de ressources est en réalité un phénomène ordinaire. Il serait donc judicieux de développer une politique de vérification de blocage qui soit efficace tout en considérant des ressources non fiables.
Cette thèse vise principalement à faire face aux limitations mentionnées ci-dessus. Nos principales contributions à la fois théoriques et algorithmiques sont les suivantes.
Premièrement, après avoir revisité les conditions de contrôlabilité des siphons (cs-propriété) et précisé les limitations des max, max′ et max′′-cs- propriétés, nous définissons la max*-cs-propriété et nous démontrons que cette nouvelle propriété est une condition non seulement suffisante mais aussi nécessaire pour la vivacité de la classe des GS^{3}PR (Generalized Systems of Simple Sequential Processes with Resources). Par la suite nous montrons comment le problème de la vérification de cette propriété et donc la vivacité des GS^{3}PR peut se ramener à la résolution d'un programme linéaire en nombre entiers.
Dans une seconde partie, nous proposons une classe de réseaux de Petri appelée M-Nets dotée d'une forte capacité de modélisation des systèmes de production automatisés. En combinant la théorie du contrôle siphon avec la théorie des régions, nous développons une méthode de prévention de blocage ayant un bon compromis entre l'optimalité du comportement et la complexité de calcul. De plus, nous proposons une méthode de synthèse d'un contrôleur maximal permissif pour une sous-classe de réseaux notée β-nets basée sur des distributions de jetons dans les siphons et évitant la génération du graphe d'accessibilité et l'énumération des siphons minimaux.
Enfin, nous proposons une méthode de conception d'un superviseur considérant les ressources nom fiables particulièrement efficace pour les systèmes pouvant être modélisés par les réseaux S^{3}PR.

Mots clés :
Systèmes de production automatisés, réseaux de Petri, blocage, siphon, synthèse de contrôleurs.

Gaiyun LIU

Supervisor Synthesis for Automated Manufacturing Systems Based on Structure Theory of Petri Nets

Abstract :
Because of automation and reduction of the number of operators, the control of industrial systems is becoming a critical issue. For automated manufacturing systems (AMS) where resource sharing is preponderant, the notion of partial or total blocking is frequent and validation before implementation is preferable to reduce the risks.
Due to the easy and concise description of the concurrent execution of processes and the resource sharing by Petri nets, many methods to verify deadlock-freeness and to synthesize controllers using structural theory or reachability graph have been proposed over the past two decades. Traditionally, a deadlock control policy can be evaluated by three performance criteria : structural complexity, behavioral permissiveness, and computational complexity. Generally, deadlock control policies based on the state space analysis can approach the maximal permissive behavior, but suffer from the state explosion problem. On the contrary deadlock control policies based on the structural analysis of Petri nets avoid in general the state explosion problem successfully, but cannot lead to the maximally or near maximally permissive controller. Moreover, the current deadlock control theory based on siphons is fairly mature for ordinary Petri nets, while for generalized Petri nets, it is presently at an early stage. On the other hand, most deadlock control policies based on Petri nets for AMS proceed on the premise that the resources in a system under consideration are reliable. Actually, resource failures are inevitable and common in most AMS, which may also cause processes to halt. Therefore, it is judicious to develop an effective and robust deadlock control policy considering unreliable resources.
This thesis aims to cope with the limitations mentioned above. Our main theoretical and algorithmic contributions are introduced as the following.
Firstly, after revisiting the controllability conditions of siphons and limitations of max, max′, and max′′-controlled siphon properties (cs-properties), we define the max*-cs-property and prove that this new cs-property is not only sufficient but also a necessary liveness condition for generalized systems of simple sequential processes with resources (GS^{3}PR). Moreover, we show the verification of this property and hence liveness of GS^{3}PR nets can be translated into resolution of an integer programming (IP) model.
Secondly, we propose a class of manufacturing-oriented Petri nets, M-nets for short, with strong modeling capability. Combining siphon control and the theory of regions, we develop a deadlock prevention method that makes a good trade-off between behavioral optimality and computational tractability. Moreover, this thesis proposes a maximally permissive control policy for a subclass of Petri nets (called β-nets) based on the token distribution patterns of siphons and avoiding the generation of reachability graphs and enumeration of minimal siphons.
Finally, we propose a design method of robust liveness-enforcing supervisors for AMS with unreliable resources. The proposed method is appropriate in particular for plants which can be modeled by systems of simple sequential processes with resources (S^{3}PR).

Keywords :
Automated manufacturing system, Petri net, deadlock, siphon, controller synthesis.