Deadlock Control of Automated Manufacturing Systems Based on Petri Nets—A Literature Review

ZhiWu Li, *Senior Member, IEEE*, NaiQi Wu, *Senior Member, IEEE*, and MengChu Zhou, *Fellow, IEEE*

***Abstract*—Deadlocks are a rather undesirable situation in a highly automated flexible manufacturing system. Their occurrences often deteriorate the utilization of resources and may lead to catastrophic results in safety-critical systems. Graph theory, automata, and Petri nets are three important mathematical tools to handle deadlock problems in resource allocation systems. Particularly, Petri nets are considered as a popular formalism because of their inherent characteristics. They received much attention over the past decades to deal with deadlock problems, leading to a variety of deadlock-control policies. This study surveys the state-of-the-art deadlock-control strategies for automated manufacturing systems by reviewing the principles and techniques that are involved in preventing, avoiding, and detecting deadlocks. The focus is deadlock prevention due to its large and continuing stream of efforts. A control strategy is evaluated in terms of computational complexity, behavioral permissiveness, and structural complexity of its deadlock-free supervisor. This study provides readers with a conglomeration of the updated results in this area and facilitates engineers in finding a suitable approach for their industrial scenarios. Future research directions are finally discussed.**

***Index Terms*—Deadlock avoidance, deadlock prevention, discrete-event system, flexible manufacturing system (FMS), Petri net.**

I. Introduction

POPULARIZED by Henry Ford, who is the founder of the Ford Motor Company, in the early of the 20th century, mass production through the use of the assembly line technique greatly contributed to the development and progress of human society and civilization. However, over the past three decades, it was challenged by quick market changes of multiple product types with a small batch. The survivability of a manufacturing system to a large extent depends on its capability to swiftly respond to the variable market requirements. Such a technological edge can be achieved through state-of-the-art manufacturing technologies, leading to the emergence and development of flexible manufacturing systems (FMS) [14], [55]. An FMS is a conglomeration of computer numerically controlled machine tools, buffers, fixtures, robots, automated guided vehicles (AGV), and other material-handling devices. It usually exhibits a high degree of resource sharing in order to increase flexibility such that manufacturers can respond to market changes quickly. The existence of resource sharing may lead to circular-wait conditions, which is the real cause of deadlocks in which each of two or more jobs in a set keeps waiting indefinitely for the other jobs in the set to relinquish resources that they hold. In such a system, once deadlocks occur, they persist and would not be resolved without the intervention from human beings or other external agency.

With the extensive applications of information technology to contemporary manufacturing systems, their safety, reliability, and some other miscellaneous requirements are met by control software [32]. Actually, modern automated manufacturing systems are increasingly becoming software-intensive systems. Compared with their establishment and maintenance, the design of control software is a complex, intricate, and arduous task for control engineers by whom deadlock problems must be carefully considered and appropriately handled, since deadlocks can not only lead to the stoppage of a part or the whole of a system, but also give rise to catastrophic results in highly automated systems such as semiconductor manufacturing and safety-critical distributed databases. Deadlock problems in automated manufacturing systems have received more and more attention from both academic and industrial communities.

The deadlock issues are originated from resource allocation systems, which could be traced as far back as to the memory assignments in an operating system by computer scientists in the 1960s [42], [85], [87], [93], [115], [181], [186], [211]. As a logical problem, deadlocks can arise in different contexts. Although a variety of deadlock control methods are developed for deadlock problems in operating systems, multiprocessing, and distributed databases, they cannot directly be applied to manufacturing systems due to different technical backgrounds. For example, as resources in a computer operating system, memory is preemptable and can be taken away from the process owning it. While resources in a manufacturing system are usually nonpreemptable. Deadlocks in a resource allocation system are in general considered to be a result of 1) deficiency of system resources; 2) an inappropriate execution order of processes; and 3) improper resource allocation logic. Summarily, there are four

Manuscript received September 21, 2010; revised February 28, 2011; accepted May 2, 2011. Date of publication October 6, 2011; date of current version June 13, 2012. This work was supported by the National Natural Science Foundation of China under Grants 61074035, 61050110145, and 61034004, the National Research Foundation for the Doctoral Program of Higher Education, the Ministry of Education, China, under Grant 20090203110009, Fundamental Research Funds for the Central Universities under Grant JY10000904001, and the Ministry of Education, and the Research Fellowship for International Young Scientists, National Natural Science Foundation of China under Grant 61050110145. This paper was recommended by Associate Editor J. Wang.

Z. Li is with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China, and also with the FARCAMT, Industrial Engineering Department, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia (e-mail: zhwli@xidian.edu.cn).

N. Wu is with the Department of Mechatronics Engineering, Guangdong University of Technology, Guangzhou 510090, China (e-mail: nqwu@gdut.edu.cn).

M. Zhou is with The Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 200092, China, and also with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: zhou@njit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCC.2011.2160626

necessary conditions for a deadlock to occur, which are known as the Coffman conditions [42].

1) *Mutual exclusion condition*: A resource cannot be used by more than one process at a time.
2) *Hold and wait condition:* Processes that already hold resources may request new resources.
3) *No preemption condition:* No resource can be forcibly removed from a process that holds it, and 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.

It is shown that the first three conditions are decided by the physical characteristics of a system and its resources. That is to say, for a system with a given set of resources, the first three conditions are either true or false. They are time invariant. However, the fourth one can be enforced to vary depending on request, allocation, and release of the resources in the system. 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.

Breaking the mutual exclusion condition implies that no process can exclusively access a resource. This mechanism is impossible for a resource that cannot be spooled. A manufacturing resource, e.g., a machine tool or a robot, cannot usually be spooled.

Removing the hold and wait condition can be achieved by requiring a process to request and be allocated all resources that are needed before its execution. Alternatively, a process is allowed to request resources only when it does not hold any resources. The former mechanism is too conservative and often leads to unacceptable low resource utilization, while the latter is generally impractical in manufacturing systems.

The preemptability of a resource depends on its nature. If a process that is holding some resources requests another resource that cannot be immediately allocated to it, then all the resources currently being held are released. However, this mechanism is usually unrealistic, difficult, or even impossible in manufacturing practice.

Circular-wait prevention allows processes to wait for resources but ensures that the waiting is not circular. The common and popular circular-wait avoidance mechanism is to establish offline a precedence to each resource and force processes to request resources in an order of increasing precedence. This forces resource allocation to follow a particular and noncircular ordering. Hence, a circular wait cannot occur.

Before the emergence of agile and flexible manufacturing modes, the occurrences of deadlocks in a system are infrequent, since most manufacturing resources are nonshared. Once a deadlock occurs, it can be solved by the intervention of operators through aborting one or more processes that are involved in the deadlock. Such a strategy sounds brute force but can lead to high system throughput since deadlocks are usually local and do not result in a global stoppage of a system. An FMS, however, aims to provide a tradeoff solution between productivity of mass production and flexibility for multiple part types. Resource sharing is an inherent and elementary feature. A deadlock occurrence degrades a system's performance. Deadlock freedom is also a key requirement for safety-critical systems. For example, in a highly automated semiconductor manufacturing system, the amount of time that wafers stay at a chamber is absolutely crucial. The occurrence of a deadlock extends the sojourn of wafers at chambers, possibly making all the wafers in such chambers scrapped. As a result, it is essential to ensure that deadlock will never 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 a resource allocation system. The physical characteristics of an FMS show that the first three deadlock conditions always hold and the only feasible doorway to eliminate deadlocks is to falsify the circular-wait condition [72].

Digraphs, automata, and Petri nets are three major mathematical tools to investigate deadlock problems in FMS [72]. Graph theory or a digraph is 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 [39], [131], [256], [257], and Fanti [63], [65]–[71]. Supervisory control theory (SCT) [199] that is based on formal languages and finite automata, originated by Ramadge and Wonham, aims to provide a comprehensive and structural treatment of discrete-event systems. As an important paradigm, SCT has a profound influence on the supervisory control of FMS under other formalisms such as Petri nets. A number of effective yet computationally efficient deadlock-control policies are developed that are based on automata. Lawley, Reveliotis, and Ferreira are distinguished experts in this area [135]–[143], [203], [262]. In particular, a theoretically significant deadlock-avoidance policy with polynomial-complexity is developed for a class of resource allocation systems in [204], which is then described in a Petri net formalism [191].

The publications on deadlock control of FMS indicate that Petri nets are increasingly becoming an important, popular, and fully fledged mathematical model to investigate the modeling and control of discrete-event systems [128], [220], [253], [254] including FMS. For example, the bibliometric analysis that is based on the Engineering Index database [56] indicates that more and more studies employ Petri nets as a tool to handle deadlock problems. This study, hence, focuses on the review of deadlock-control approaches that are based on them.

The rest of this paper is organized as follows. Section II outlines deadlock-handling strategies, whose merits and drawbacks are briefly discussed in an FMS context. Approaches to deadlock detection and recovery, deadlock avoidance, and deadlock prevention are reviewed in Sections III, IV, and V, respectively. Section VI proposes open problems in this area. Section VII concludes this paper, and Section VIII presents bibliography notes.

II. Deadlock-handling Strategies

From a conceptual standpoint, there are four strategies to handle deadlocks in automated manufacturing systems: deadlock ignoring, prevention, avoidance, and detection and recovery.

Deadlock ignoring, which is known as the Ostrich algorithm [97], is employed in a resource allocation system if the probability of deadlocks is tiny and the enforcement of other deadlock-control strategies is technically or financially difficult. A notable example of deadlock ignoring is in UNIX, where a deadlock can occur if the process table is full and all the processes still attempt to fork more subprocesses. This situation, however, appears rather infrequently, and to prevent this scenario needs cumbersome restrictions, which is usually ignored. In theory, a deadlock-ignoring strategy is a tradeoff between convenience and correctness. In an FMS, deadlock ignoring is feasible and reasonable from the technical and economic points of view if the degree of resource sharing is low. This strategy is widely used in the early development stage of FMS.

Deadlock prevention is considered to be a well-defined problem in resource allocation studies. It is usually achieved by using an offline computational mechanism to control the request for resources to ensure that deadlocks never occur. In other words, resources are granted to requesting processes in such a way that a request for a resource never leads to a deadlock. The goal of a deadlock-prevention approach is to impose constraints on a system's evolution to prevent it from reaching deadlock states. In this case, the computation is carried out offline in a static way and once a control policy is established, the system can no longer reach undesirable deadlock states. The simplest way to prevent a deadlock is to acquire all the needed resources before a process starts its execution. This is highly inefficient since it decreases system concurrency and stifles its operational flexibility. A major advantage of deadlock-prevention algorithms is that they require no runtime cost since problems are solved in system design and planning stages. The major criticism is that they tend to be too conservative, thereby reducing the resource utilization and system productivity. Deadlock prevention is usually considered to be applicable to safety-critical systems in which deadlocks may lead to serious results and cause enormous economic loss.

In a deadlock-avoidance strategy, a resource is granted to a process only if the resulting state is safe. A state is called safe if there exists at least one execution sequence that allows all processes to run to completion. In order to decide whether the forthcoming state is safe if a resource is allocated to a process, every cell controller and the global controller in an FMS need to keep track of the global system state. This means that huge storage and an extensive communication ability are necessary. This is also true for the case that an FMS operates under control of a distributed control system in which there is no global controller. Furthermore, it is computationally expensive to check the safety of a state due to a large number of reachable states in a real-world system. It is worthy to note that too aggressive deadlock-avoidance methods usually lead to higher resource utilization and throughput, but do not totally eliminate all deadlocks for some cases. In such cases if a deadlock arises, suitable recovery strategies are still required [131], [235], [257]. Conservative methods eliminate all unsafe states and deadlocks, and often some legal and acceptable states, thereby degrading the system performance. On the other hand, they are intended to be easy to implement.

In deadlock detection and recovery, however, resources are granted to a process without any check. The status of resource allocation and requests are examined periodically to determine whether a set of processes is deadlocked. This examination is performed by a deadlock-detection algorithm. If a deadlock is found, the system recovers from it by aborting one or more deadlocked processes. A primary requirement for a correct deadlock-detection algorithm is that it must detect all possible deadlocks and does not report nonexistent deadlocks. In manufacturing practice, human operators are often needed for this strategy and, thus, can be very expensive.

The persistence of a deadlock means the stoppage of a partial or whole system. Deadlock recovery is by no means a trivial task. In fact, the problem of promptly and efficiently resolving a detected deadlock is as important as the deadlock detection itself.

A deadlock is resolved by aborting one or more processes that are involved in the deadlock and granting the released resources to other processes that are involved in the deadlock. A process is aborted by withdrawing all its resource requests, restoring its current state to an appropriate previous state, relinquishing all the resources that it has acquired after that state, and restoring all the relinquished resources to their original states. In the simplest case, a process is aborted by starting it afresh and relinquishing all the resources that it holds [211]. As stated previously, a deadlock is resolved by aborting at least one or more processes that are involved in the deadlock and granting the released resources to other processes that are involved in the deadlock. Usually, a deadlock resolution involves the following steps that are computationally expensive.

1) Select a victim, a process to be aborted, for the optimal resolution of a deadlock.
2) Abort the victim, release all the resources held by it, restore all the released resources to their previous states, and grant the released resources to deadlocked processes.

In an FMS, if the cost of aborting a process that is involved in a deadlock is high, technically difficult, or impossible, a feasible alternative is the usage of spare resources that can function as those involved in deadlocked processes. This strategy, however, implies extra equipment investment.

The suitability of a deadlock-handling strategy to a large extent depends on the application cases. Both prevention and avoidance are conservative. They are usually used if the deadlocks are frequent or their occurrences lead to a serious result. In contrast, deadlock detection and recovery is an optimistic strategy that grants a resource to a request as long as the resource is available. Hopefully, this resource allocation does not lead to a deadlock. As a result, this strategy is suitable for the system in which deadlocks are rare, their occurrences do not lead to severe results, and their recovery is technically and financially affordable.

III. Deadlock Detection and Recovery

The stability of deadlocks is shown by their persistence. Once a deadlock occurs, it persists forever. As a result, it is not difficult to detect deadlocks. For example, the existence of a cycle in the resource allocation graph of a system is a necessary condition for deadlocks. An algorithm to detect a cycle in a graph requires an order of n^2 operations, where n is the number of vertices in the graph, i.e., the number of resources in general manufacturing practice.

Deadlocks in an FMS can be detected by a prior simulation through executing the most favored processing sequences as long as possible in order to determine the most probable processes that are involved in a deadlock as well as the resources held by these permanently blocked processes. In theory, such an approach, just as if one stands by a tree stump waiting for a hare, cannot completely disclose all deadlock scenarios in a system. Formal tools to detect deadlocks include resource allocation graphs, Petri nets, and other paradigms and methods. For example, a well-known result is the *Deadlock Theorem*: S is a deadlock state if the resource allocation graph of S is not completely reducible [93]. Moreover, it is shown that the existence of a cycle in a resource allocation graph is a necessary condition for deadlocks. In some special cases, it is also sufficient. In a Petri net formalism, deadlocks are closely tied to a structural object, namely siphons. Once a siphon in the Petri net that models a system is unmarked (i.e., no tokens sojourn in it), all its related transitions, i.e., events, are dead, implying the permanent blocking of some or all processes in the system.

If a system is in a deadlock state, some recovery strategies must be applied such that the deadlock is resolved. Three kinds of actions can be taken for recovery: process termination, resource preemption, and resource reservation.

1) *Process termination*: Aborting all deadlocked processes is usually a bad bargain. Alternatively, one can kill a process at a time until the deadlock disappears, which, however, implies high overhead since a deadlock-detection algorithm is run at each step.
2) *Resource preemption:* A victim, i.e., a process that is not one involved directly in the deadlock, is carefully selected from which enough resources are preempted such that they are made available to deadlocked processes to resolve the deadlock. Thereafter, the process whose resources are preempted has to be rolled back to some previous safe state and is made to continue its operations. However, resource preemption is in general infeasible in FMS.
3) *Resource reservation:* As a similar case of the resource preemption strategy, this approach calls resources reserved beforehand for the deadlocked processes such that the deadlock can be eliminated.

In an automated FMS, it is hard to imagine that the aforementioned three strategies can be implemented automatically. That is to say, a deadlock in it must be, in general, resolved through manual intervention whatever strategies are employed, which implies high cost.

Deadlock recovery can be considered as a special case of error recovery that has been extensively investigated over the past years [187], [188], [200], [235], [236], [274]. A graph-theoretical approach is introduced in [64] for deadlock detection and recovery in FMS whose resources have multiple capacities. The concept of maximal-weight zero-outdegree strong components is used to characterize and detect deadlocks. Once a deadlock is detected, an automatic procedure is initialized by using a reserved central buffer with unit capacity to resolve the deadlock. Other detection and recovery approaches that are based on graph theory can be found in [129], [131], [144], [256], [257], and [267].

There is no much work on deadlock detection and recovery that is based on a Petri net formalism. Basile *et al.* [13] consider the supervisory control problem for a Petri net whose structure is known and its initial marking is fully or partially unknown. Markings are estimated by observing the occurrences of events that are assumed to be controllable. The inaccurate marking estimate can, in theory, lead to deadlocks that are called observer-induced deadlocks, which is decided by checking whether a particular deadlock condition is satisfied. When the controlled system satisfies the condition and, consequently, no further event is observed for a sufficiently long time, then an automatic deadlock-recovery procedure is awakened. Similar ideas can be found in [62], and the related results on marking estimation and deadlock recovery are summarized in [80].

IV. Deadlock Avoidance

The work in [235] is usually considered to be earlier and seminal research that demonstrates the use of Petri nets in modeling and deadlock control of FMS. More importantly, deadlock prevention and avoidance approaches are proposed. The former is achieved by an exhaustive path analysis of the Petri net model of the given FMS. It is feasible for a system whose state space is small enough to fit in the memory of our computer. Deadlock avoidance is effectively implemented by an online monitoring and control system in which the Petri net model is used to look ahead into the future system evolution in order to make a correct deadlock-free resource allocation strategy. It is shown that deadlock avoidance is feasible for large real-world systems. However, deadlock states cannot be completely eliminated, in general, because of the limited number of look ahead steps.

Motivated by the banker's algorithm, i.e., a resource allocation and deadlock-avoidance algorithm [52] for operating systems, the work in [60] proposes a deadlock-avoidance algorithm for a class of manufacturing-oriented Petri nets, which is called S*PR, denoted by $N = (P_0 \cup P_A \cup P_R, T, F)$, where P_0, P_A, P_R, and $T = \cup_{i=1}^{n} T_i$ are the sets of idle places, activity (operation) places, resources, and transitions, respectively. $F \subseteq (P \times T) \cup (T \times P)$ is the flow relation between $P = P_0 \cup P_A \cup P_R$ and T. The complexity of the deadlock-avoidance algorithm is $O(|P_A|^2 |P_R| k)$, where $k = \max\{|T_i| \| i \in \{1, 2, \ldots, n\}\}$. The algorithm is then used to a more general class of Petri nets, which are called nonsequential resource-allocation processes [61].

Wu and Zhou have been active in the area of deadlock avoidance by using resource-oriented Petri nets (ROPN) as a formalism [255]. The main contributions in [242] consist of a new class of Petri nets, which is called colored resource-oriented Petri nets (CROPN), and necessary and sufficient conditions for deadlock-free operations from which a maximally permissive control law is derived. The deadlock freedom of interactive subnets in a CROPN is verified by the control law. The whole CROPN of the system is shown to be live if each of its interactive subnets is controlled.

The work in [243] addresses the issue whether the maximally permissive policy proposed in [242] is optimal in terms of productivity in an environment where dispatching rules dominate and optimal scheduling is impossible due to either unaffordable computation required or changing operational parameters and structures. A novel deadlock-control policy is developed such that it can avoid deadlock completely, and reduce starvation and blocking situations significantly by releasing an appropriate number of jobs into the system and controlling the order of resource usage based on state information in the net model. By using CROPN, the work in [245] handles the modeling, deadlock avoidance, and conflict resolution in AGV systems with bidirectional and unidirectional paths. The contributions in [246] are twofold. One is the system modeling by integrating the models of the AGV system and part processing processes through macro transitions, where CROPN are used as a modeling paradigm. The other is the development of a deadlock-avoidance policy. The policy is shown to be maximally permissive with computational complexity of $O(n^2)$ where n is the number of machines in the system. The work in [247] aims to find shortest time routes for bidirectional AGV, while both deadlock and blocking are avoided. Compared with other methods, the proposed one can offer better solutions. Also, its performance analysis is conducted via experimental studies.

Different from the traditional modeling approach for robots in an automated FMS, the study in [248] presents a novel way to model robots. Under such a modeling paradigm, an interesting and appealing result is obtained, i.e., robots have no contribution to deadlocks. Instead, they can be used to resolve deadlocks by also serving as temporary part storage devices. A new deadlock-control policy is proposed by treating robots as both material-handling devices and buffers, which is more permissive than the existing ones.

This interesting and attractive proclamation that is made in [248], however, deserves to be carefully considered. It is known that a net model is a mathematical and formal description of a system. To fully understand a system, it is required that the model must sufficiently describe the physical characteristics of the system. Whether resources can contribute to potential deadlocks in an FMS is completely decided by the FMS's configuration, part routings, and a resource allocation policy, and should be independent from the modeling styles and skills. It is reasonable that such a robot modeling treatment can facilitate the development of a deadlock-avoidance policy. However, solid evidence that the derived policy is definitely more permissive than the existing ones [59], [148] remains to be found.

It seems that such a modeling paradigm condenses the states in the traditional *process-oriented* models [251], [274] that are used in most deadlock-handling work [59], [148], [271]–[276]. Moreover, the modeling treatment of robots in [248] excludes some behavior of the plant, which is inadmissable from the deadlock-control point of view. For example, a robot is not allowed to pick up a raw part if its serving machine tool has no capacity to host the part. Actually, some control ideas are mingled with the modeling in [248].

Behavioral permissiveness is one of the most important criteria to evaluate a deadlock-free supervisor for an FMS. For a class of Petri nets, i.e., Systems of Simple Sequential Processes with Resources (S^3PR), Xing *et al.* [260] propose an optimal deadlock-avoidance policy with polynomial complexity when an S^3PR satisfies a special condition with respect to its net structure and initial marking. Resource-transition circuits (RT circuits), perfect RT circuits (PRT circuits), and maximal PRT circuits (MPRT circuits) are defined. A resource is said to be a ξ-resource if it is shared by two MPRT circuits that do not contain each other. It is claimed that in an S^3PR without ξ-resources, the reachable markings are either safe or deadlocks. As a result, a one-step look-ahead method can be utilized to avoid deadlocks completely. It is shown that the supervisor for an S^3PR without ξ-resources is maximally permissive, i.e., all deadlock states are eliminated and all safe states are kept in the controlled net system. In the case of the existence of ξ-resources, a suboptimal supervisor can be obtained.

The work in [260] is extended by Wu and Zhou [252]. They consider sequential resource allocation problems in FMS in which each processing step needs a single type of resources. They propose necessary and sufficient conditions under which a one-step look-ahead maximally permissive deadlock-avoidance policy exists. Also, an approach is developed to derive such a policy if it exists. Note that in [252], a system is modeled with ROPN, and in [260] a system is modeled with traditional *process-oriented* Petri nets. The former is claimed to be more compact and effective for deadlock resolution [251] than the latter. Similar results can also be found in [67], [142], and [207] in which automata or digraph are used.

The work in [3] contributes a hybrid approach that combines deadlock prevention and avoidance for a class of Petri nets, i.e., System of Sequential Systems with Shared Resources (S^4R), which is more general than S^3PR and allows multiple resource requirements in performing an operation on a part. When the deadlock-prevention phase cannot eliminate deadlocks, an online dynamic resource allocation policy is applied to the controlled system in order to minimize the probability of deadlock occurrences by trying to make always marked the monitors that results from the deadlock prevention. Even if both deadlock prevention and avoidance are employed, the occurrences of deadlocks remain possible in theory. Hence, this method is not feasible in highly automated FMS. The idea in [3] can be improved by considering a polynomial deadlock-prevention policy that is developed by Park and Reveliotis [191].

The deadlock-avoidance algorithms that are reported by Roszkowska and colleagues in [6], [7], [209], and [210] are usually considered to be classical and seminal work in deadlock control by using Petri nets. Their contributions can be used for a class of Petri nets called Production Petri Nets (PPN) that are less general than S^3PR proposed in [59]. The policy uses a feedback mechanism by monitoring the system state concerning currently active jobs to make a correct resource allocation decision by disabling some enabled transitions. It is shown that the policy is not optimal, and in some cases, it is overly conservative.

Hsieh and Chang [95] propose a bottom-up method to synthesize a controlled production Petri Net (CPPN) model of FMS, which consists of resource subnets, job subnets, and exogenous controls to describe the deadlock-avoidance problem. Resource and job subnets are merged into a PPN that models the interactions among operations, resources, and jobs in FMS. Every transition is associated with a control place that controls its firing even if it is enabled in the sense of traditional net theory. A CPPN is decomposed into controlled production subnets of individual job types, and a necessary and sufficient condition is developed for it to be live, which is motivated by the concept of minimal resource requirements at the idle state places to complete all the jobs in the CPPN. A sufficient validity test procedure with polynomial complexity is proposed to check whether the execution of a control action is valid to maintain the liveness of the CPPN, and a synthesis algorithm is developed to determine a sequence of valid control actions to carry out the given dispatching policy. It is cogent that the proposed deadlock-avoidance controller can achieve high resource utilization.

Machine and device failure is inherent and common in many FMS. The failure of a single device such as a workstation or even an individual sensor can cause the whole system to shutdown. Based on controlled assembly Petri nets (CAPN), Hsieh [96] develops a suboptimal deadlock-avoidance algorithm with polynomial complexity for manufacturing assembly processes with unreliable resources. Resource failure is modeled as loss of tokens in a Petri net model to represent the unavailability of resources in the course of recovery procedures. Three types of token losses are defined to model resource failures in 1) a single operation; 2) multiple operations of a production process; and 3) multiple operations of multiple production processes. Sufficient conditions are established for each type of token loss, which guarantee the liveness of a CAPN after some tokens are removed. An algorithm is proposed to conduct feasibility analysis by searching for recovery control sequences and to keep as many types of production processes as possible in production.

The significant contributions to the area of deadlock avoidance for FMS by using Petri nets as a formalism include [242]–[252], while Lawley, Reveliotis, Ferreira, and Fanti dominate the area of automata and graph theory to deal with deadlock control [53], [63]–[73], [135]–[143], [208].

It is shown that optimal deadlock avoidance is NP-complete even for Liner Single-unit Resource Allocation Systems that are equivalent to PPN [207]. However, for Disjunctive Single-unit Resource Allocation Systems (DIS-SU-RAS), if the capacity of each resource is bigger than one, its optimal supervisory control can be done in polynomial time. A DIS-SU-RAS is actually equivalent to S^3PR.

V. Deadlock Prevention

Deadlock prevention for FMS has been investigated extensively, leading to a vast stock of results. This section reviews typical deadlock-prevention strategies that are, by using Petri nets as a formalism, developed on the basis of different techniques, i.e., initial marking configuration, partial or complete reachability graph analysis, structural analysis, and their combination.

A. Initial Marking Configuration

The seminal work in [271] formulates a Petri net synthesis method of modeling automated manufacturing systems with shared resources. The concepts of parallel and sequential mutual exclusions are proposed. First, the places in a Petri net model of an automated manufacturing system are categorized into three classes: A, B, and C. An A-place is also called an activity place, representing a processing step of a raw part. It is unmarked at an initial marking, implying that no processing steps are activated in the initial state of a manufacturing system. B-places are called fixed resource places that are used to model manufacturing activity executors such as machine tools, robots, buffers, and part conveyors. A C-place is called a variable resource place that represents the availability of raw parts, fixtures, pallets, etc. At an initial marking, the number of tokens in a B- or C-place is greater than zero. Based on the place partition, Zhou and DiCesare find a relationship between the markings of B- and C-places, under which the Petri net model is live, bounded, and reversible.

In a decade following the work in [271], an enormous amount of work has been done by DiCesare, Jeng, and Zhou [116]–[118], [120], [123], [124], [272]– [274]. Liveness conditions are established for a variety of manufacturing-oriented Petri net subclasses such as Process Nets with Resources (PNR) [122], Resource Control Nets (RCN)-merged Nets [119], and Extended ECN (ERCN)-merged nets [258], which are usually expressed as the relationships between the initial markings of B-places and C-places. The systematic modeling techniques of Petri nets for automated manufacturing systems are gradually shaped. Up to now, Petri net modeling theory has basically become mature although some work was reported in recent years [17], [249]. In summary, the research work in the early stages of this direction combines modeling and control such that the resulting net model has some desired behavioral properties such as liveness, boundedness, and reversibility. The Petri nets that are considered in Zhou and Jeng's work are usually ordinary, i.e., the weight of an arc is one.

Fig. 1(a) shows a small manufacturing system whose Petri net model is depicted in Fig. 1(b), where p_1 is a C-place, p_5 and p_6 are B-places, and the others are A-places. It is verified that the net model in Fig. 1(b) is live if $M_0(p_5) + M_0(p_6) > M_0(p_1)$. The initial marking configuration for liveness is in some cases rather conservative and deteriorates the performance of the controlled system. Consider the net shown in Fig. 2(a) that is extended by adding one processing stage with a machine tool of capacity four prior to the robotic cell in Fig. 1(a). In order to keep liveness, $M_0(p_1) \leq 2$ is necessary if $M_0(p_5) = 2$ and $M_0(p_6) = 1$. In this case, the resource that is represented by p_8 cannot be fully utilized. This approach tackles the liveness problem by a way of global restriction, not focusing on a part or component that results in deadlocks. Furthermore, deadlocks cannot be eliminated in some systems by configuring an initial marking. For the net shown in Fig. 2(b), one cannot find an initial marking at which the net is live. However, its liveness can be enforced by properly designing a monitor ν, as depicted in Fig. 2(c).

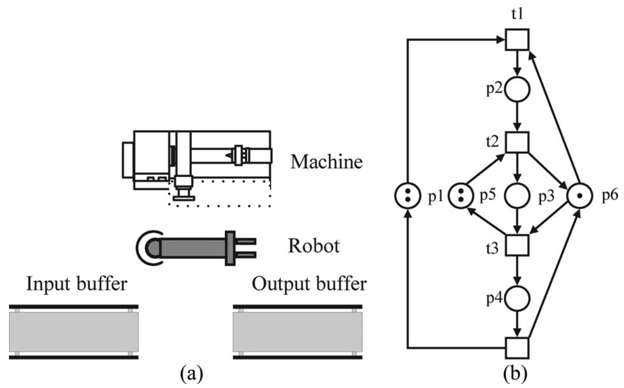

Fig. 1. (a) Manufacturing system and (b) its net model (N, M_0).

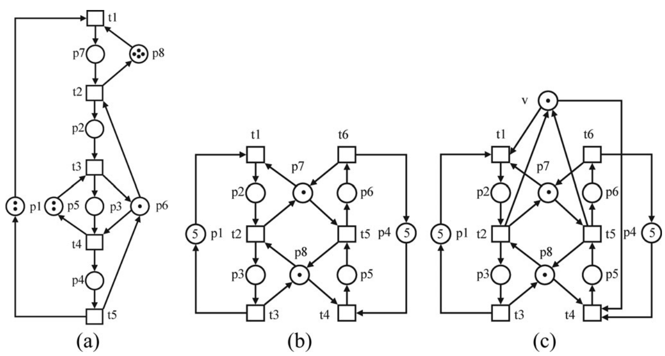

Fig. 2. (a) Net model (N_1, M_1), (b) a net model (N_2, M_2), and (c) a controlled system (N_3, M_3) for (N_2, M_2).

Recent contributions in [11], [12], and [281] belong to this class of research mentioned earlier. The work in [12] deals with the verification of *soundness*, i.e., correct termination, of a procedure in workflow nets [1]. The soundness of a workflow net is tied to its liveness and boundedness. Based on workflow nets, a novel class of Petri nets, which is called workflow nets with resources, is proposed. A workflow net with resources can model rather general manufacturing systems with assembly and disassembly operations. Variations of a workflow net with resources include G-tasks [11] and G-systems [281]. Among the contributions by Barkaoui and his colleagues, one of the most important results is the controllability condition of a siphon, which is sufficient for the absence of dead transitions that are related to it.

Let $S \subseteq P$ be a subset of places in a Petri net $N = (P, T, F, W)$ and $x \in P \cup T$ be a node. ${}^\bullet x = \{y|(y, x) \in F\}$ and $x^\bullet = \{y|(x, y) \in F\}$ are called the preset and postset of x, respectively. ${}^\bullet S = \cup_{x \in S} {}^\bullet x$ and $S^\bullet = \cup_{x \in S} x^\bullet$ are called the preset and postset of S, respectively. S is called a siphon (trap) if ${}^\bullet S \subseteq S^\bullet$ ($S^\bullet \subseteq {}^\bullet S$). A siphon is said to be minimal if it contains no other siphons as its proper subset. A minimal siphon is strict if it contains no trap. The controllability of a siphon in a G-system is demonstrated by the following example.

Let $N = (\{i, o\} \cup P_A \cup P_R, T, F, W)$ be a G-system, where i is called a source place with ${}^\bullet i = \emptyset$, o is called a sink place with $o^\bullet = \emptyset$, P_A is a set of activity places, and P_R is a set of resource places. Let $P = \{i, o\} \cup P_A \cup P_R$. F is the flow relation between nodes in $P \cup T$, and W assigns a nonnegative integer weight to any pair (x, y) with $x, y \in P \cup T$. $\forall r \in P_R$, there exists a minimal P-semiflow $f_r : P \to \mathbb{N}$ with $f_r(r) = 1$ and $\forall p \in P_A$, there exists a minimal P-semiflow $f_p : P \to \mathbb{N}$, where $\mathbb{N}$ is a set of nonnegative integers. Let S be a minimal siphon, $g_S = \sum_{r \in S \cap P_R} f_r$, $\text{Out}(S) = \|g_S\| \setminus S$, $h_S = \sum_{p \in \text{Out}(S)} f_p$, $d_S = \max_{p \in \text{Out}(S) \cap \|h_S\|} g_S(p)$, and $z_S = g_S - d_S h_S$.

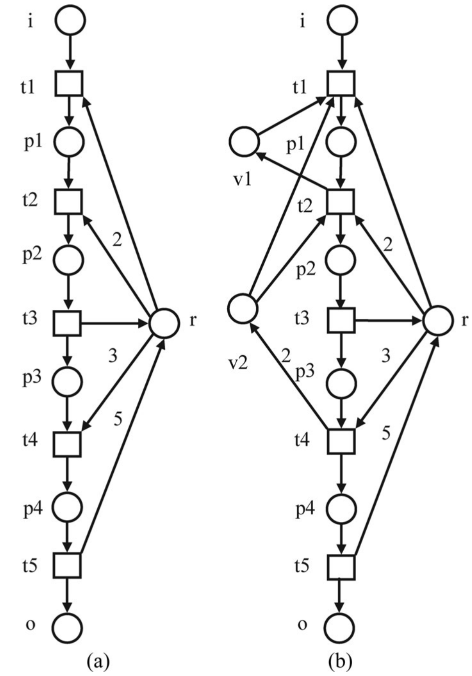

Fig. 3. (a) G-system (N, M_0) and (b) its controlled net system (N_1, M_1).

Siphon S is max controlled if

$$z_S^T M_0 > \sum_{p \in S} z_S(p)(\max_{p^\bullet} - 1)$$

where $\|g_S\| = \{p | p \in P, g_S(p) \neq 0\}$ is called the support of a $|P|$-vector g_S, and $\max_{p^\bullet} = \max\{W(p, t) | t \in p^\bullet\}$ is the maximal weight of the arcs from p to the transitions in its postset.

Take the net that is shown in Fig. 3(a) as an example, where we assume that $M_0 = j.i + k.r$, implying that source place i has j tokens, resource place r has k tokens, and the others are unmarked at the initial marking. Note that $M_0 = j.i + k.r$ is a multiset representation of vector M_0. According to the definition of G-systems, we should have $k \geq \max\{f_r(p) | f_r(p) \neq 0\} = 5$. It is verified that $S = \{r, p_2, p_4\}$ is a strict minimal siphon. We, hence, have

$$g_S = r + p_1 + 3p_2 + 2p_3 + 5p_4, \text{Out}(S) = \|g_S\| \setminus S = \{p_1, p_3\}$$

$$h_S = \sum\nolimits_{p \in \text{Out}(S)} f_p = i + p_1 + p_2 + p_3 + p_4 + o$$

$$d_S = \max_{p \in \text{Out}(S) \cap \|h_S\|} g_S(p) = \max\{g_S(p_1), g_S(p_3)\} = 2$$

$$\begin{aligned} z_S = g_S - d_S h_S &= (r + p_1 + 3p_2 + 2p_3 + 5p_4) \\ &\quad - 2(i + p_1 + p_2 + p_3 + p_4 + o) \\ &= r + p_2 + 3p_4 - p_1 - 2i - 2o. \end{aligned}$$

Considering that $M_0 = j.i + k.r$, i.e., $\forall p \notin \{i, r\}$, $M_0(p) = 0$, $\max_{r^\bullet} = 3$, $\max_{p_2^\bullet} = 1$, and $\max_{p_4^\bullet} = 1$, S is max controlled if $k - 2j > 2$. From this example, the controllability of a siphon can be represented by a relationship between the initial marking of source and resource places. Clearly, $k = 5$ leads to $j = 1$ for S to be max controlled in Fig. 3(a). The unique strict minimal siphon satisfies max-controlled siphon property, controlled-siphon property (cs-property) for short, implying the liveness of the G-system. A G-system satisfies the cs-property if there exists an initial marking such that the controllability of each siphon is ensured.

The deadlock prevention based on the initial marking relationship of source and resource places is overconservative in general. A monitor-based supervisor is shown in Fig. 3(b) that is more permissive than the one in Fig. 3(a) where the cs-property is ensured by the initial marking relationships.

B. Structural-Analysis Methods

Traditionally, a Petri net supervisor is a Petri net that consists of monitors and transitions in a plant net model. A Petri net supervisor is said to enforce liveness to a plant if the controlled system that results from the composition of the supervisor and plant via shared transitions is live. In this case, it is called a liveness-enforcing Petri net supervisor, or liveness-enforcing supervisor for short.

A self-loop in a Petri net cannot be mathematically shown in its incidence matrix even if the two arcs involved in the self-loop do not have the same weights. For this reason, a computed supervisor based on algebraic operations in a computer is usually self-loop free, i.e., pure. Unless otherwise stated, a supervisor is concerned with a pure net.

The major weakness of the deadlock prevention in which liveness is conditioned by an initial marking relationship between B-places and C-places is its conservativeness. As is known, a direct and remarkable consequence is that the productivity of a system can be deteriorated. In 1995, seminal work was conducted by Ezpeleta *et al.* [59] who developed a design method of monitor-based liveness-enforcing Petri net supervisors. It is usually considered to be a classical contribution that utilizes structural-analysis techniques of Petri nets to prevent deadlocks in FMS. First, a class of Petri nets, which is called S^3PR, is proposed, and the relationship between strict minimal siphons and liveness of an S^3PR is established. It is shown that an S^3PR is live iff no siphon can be emptied. For each strict minimal siphon that can be empty at a reachable marking, a monitor is added such that it is controlled, i.e., cannot be unmarked at any reachable marking. After all siphons are controlled, the resulting net that is called a controlled net system is live.

The significance of this approach lies in the fact that it successfully separates a plant net model and its supervisor such that there is a clear boundary between them, as done in R-W theory of supervisory control for discrete-event systems [199], [241] where automata are used to be a modeling formalism. The liveness-enforcing supervisor in [59] is a Petri net that consists of the monitors and the transitions of the plant model. Note that an S^3PR can model automated manufacturing systems with flexible process routes but cannot model systems with assembly and disassembly operations, since it is composed of state machines and resources. State machines are a typical ordinary Petri net subclass in which each transition has only one input place and one output place.

Unfortunately, the approach in [59] suffers from the following problems: behavioral permissiveness, computational complexity, and structural complexity. The behavioral permissiveness problem is referred to as the fact that the permissive behavior of a plant net model is overly restricted by the deadlock-prevention policy, i.e., the supervisor excludes some safe (admissible) states. This is so since the output arcs of a monitor are led to the source transitions of the net model, which limits the number of workpieces to be released into and processed by the system. A source transition is the output transition of an idle place, which models the entry of raw parts into the system. Computational complexity results from the complete siphon enumeration that is necessary to compute a supervisor [59]. As known, the number of siphons grows fast and in the worst case grows exponentially with respect to the size of a net model. Structural complexity is referred to as the number of monitors in a liveness-enforcing supervisor, which is in theory exponential with respect to the size of a net model since every strict minimal siphon that can be unmarked at a reachable marking needs a monitor to prevent from being emptied. The structural complexity of a supervisor means extra cost in system verification, validation, and implementation. Since 1995, much work has focused on solving the aforementioned problems.

For a class of Petri nets, which is called PPN, Xing *et al.* [259] propose a deadlock-prevention policy. A transition in a PPN has only one input activity place and one input resource place. It is said to be process enabled at a marking if its input activity place is marked, and it is resource enabled if its input resource place is marked. A transition in a PPN is enabled if it is both process enabled and resource enabled. Then, the concept of deadlock structures is proposed, which is a set of transitions that are process enabled but not resource enabled. For each deadlock structure, a monitor is added such that a liveness-enforcing controlled system can be computed. In addition, the concept of critical resources is formulated. It is shown that a maximally permissive, i.e., optimal, liveness-enforcing controlled net system can be obtained if the capacity of each critical resource is greater than one. However, the supervisor in [259] also suffers from computational-complexity and structural-complexity problems. The number of deadlock structures is, in theory, exponential with respect to the size of a PPN. It can be shown that a deadlock structure can be used to derive a strict minimal siphon.

Wang *et al.* [237] show that the supremum of the strict minimal siphons in a PPN is $2^n - n - 1$, where n is the number of resource places. As a result, the deadlock-prevention policy in [259] needs in the worst case to add $2^n - n - 1$ monitors, leading to a supervisor with exponential complexity in structure with respect to the plant size. Its computational-complexity problem results from the complete enumeration of deadlock structures. Although a PPN is a subclass of an S^3PR, the idea underlying the maximally permissive deadlock-prevention policy is significant. Recently, it has been extended to design maximally permissive supervisors for S^3PR [162], [260].

Fairly speaking, both studies in [59] and [259] are of equal significance and seminality. From literature analysis, however, the work in [59] has more profound and far-reaching influence on the development of deadlock-prevention methods that are based on Petri nets as a formalism. The reasons for this are twofold. First, siphons, i.e., a well-defined and well-known structural object, are used to characterize and analyze deadlocks in [59] such that the policy is readily understandable. Xing *et al.* [259] use a newly defined concept of deadlock structures. Second, a more general class of Petri nets is defined in [59].

Because of the inherent complexity of Petri nets, any deadlock-prevention policy that depends on complete siphon enumeration suffers definitely from an exponential complexity problem with respect to the size of its plant net model [155]. 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. Chu and Xie [40] propose a deadlock-detection method by solving a mixed integer programming (MIP) problem, which is a mathematical programming implementation of the aforementioned traditional siphon solution approach. 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 [75], [240], 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.

Motivated by the fact that deadlock control is usually concerned with minimal siphons in a Petri net, Huang *et al.* and Li and Liu investigate minimal siphon extraction methods from a maximal unmarked siphon [106], [154]. A software package that can do so is developed by Liu *et al.* [168]. A minimal siphon in a structurally bounded ordinary net can be directly found by solving an MIP problem [81], [84]. Similar work on minimal siphon extraction using MIP is reported by Chao [29] and Li [145]–[147].

In order to tackle the computational-complexity problem in [59], the MIP-based deadlock-detection approach finds its further applications. Huang *et al.* develop an iterative deadlock-prevention policy for S^3PR, which consists of two phases [106]. The first is called siphon control, and the second is control-induced siphon control. At each iteration, a maximal unmarked siphon is computed by using the MIP-based deadlock-detection method. Then, a minimal siphon is derived from the maximal one. A monitor is added such that the minimal siphon is controlled by the enforcement that the set consisting of the monitor and the complementary set of the derived minimal siphon is the support of a P-semiflow. Repeat the aforementioned steps until all siphons in the original plant model are controlled. The second phase becomes necessary if the resulting net after the first phase contains deadlocks. At the second phase, a minimal siphon that contains at least a monitor is derived by solving an MIP problem. Then, a monitor is added to make the siphon controlled by leading the output arcs of the monitor to the source transitions of the plant model. This step is repeated until the MIP problem shows that there is no unmarked siphon at a reachable marking, implying that liveness is enforced. Experimental study shows that the two-phase policy is more permissive than the one in [59] and [148], although no formal proof is provided. However, there is some uncertainty in the number of reachable states of the controlled system. This is not surprising since in the second phase selection of different siphons to control can lead to liveness-enforcing controlled systems with different permissive behavior.

The two-phase deadlock-prevention policy in [106] improves the work in [59] in a number of aspects. First, a more permissive supervisor can be obtained. Second, the computational cost of a liveness-enforcing supervisor is reduced through the use of the MIP-based deadlock-detection method. However, there is no definite improvement on the structural complexity of supervisors. Furthermore, the iterative deadlock-prevention approach as usual generates redundant monitors whose removal from the final controlled system does not change its liveness property and behavior, since the addition of a monitor in an iteration step may make implicitly controlled the monitors that are added prior to the step. Later, the idea is applied to ES^3PR [107], which is a more general class of Petri nets than S^3PR.

The work in [27] proposes the concept of basic and compound siphons. First, monitors are added to each basic siphon. Then, it finds conditions for a compound siphon to be implicitly controlled. This research avoids the problem of a siphon enumeration and reduces the number of subsequent time-consuming MIP iterations.

Aiming at improving the behavioral permissiveness of a liveness-enforcing Petri net supervisor for an S^3PR, the work in [9] proposes the concept of basic and independent strict minimal siphons, which, in fact, motivates the development of elementary siphons in [148] and [161]. For each basic or independent strict minimal siphon, a monitor is designed to prevent it from being unmarked. In a general case, the addition of monitors leads to control-induced siphons whose controllability is ensured by properly configuring the initial marking of the monitors for independent siphons. A case study shows that the proposed deadlock-prevention policy is nearly optimal and more permissive than that in [59].

In order to model general systems, a variety of manufacturing-oriented Petri net subclasses are proposed such as augmented marked graphs [40], S^4R [3], S^4PR [222], Weighted System of Simple Sequential Processes with Resources (WS^3PR) [221], PNR [122], RCN-merged nets [119], ERCN-merged nets [258], ERCN*-merged nets [123], Systems of Simple Linear Sequential Processes with Resources [190], System of Simple Sequential Processes with General Resource Requirements (S^3PGR^2) [191], System of Simple Sequential Processes with Multiple Resources (S^3PMR) [108], G-tasks [11], Workflow Nets with

Shared Resources [12], S^{5}PR, SPQR [175], S*PR [60], PC^{2}R [176], and G-systems [281].

From their definitions, S^{4}R, S^{4}PR, and S^{3}PGR^2 are equivalent, which are named by different researchers. Similarly, ES^{3}PR [107] and S^{3}PMR [108] are equivalent.

Most of the extended versions from PPN [7] and S^{3}PR are generalized Petri nets in which there at least exists an arc whose weight is greater than one. Note that deadlock control in a generalized Petri net is much more difficult than that in an ordinary one. In an ordinary net, the weight of an arc is one. This implies that the transitions in the postset of a marked siphon will not be disabled totally. That is to say, there necessarily exist enabled transitions in the postset of a marked siphon. Because of this, an elegant result in an ordinary net is developed, which is invariant-controlled siphons [133]. A siphon is said to be invariant controlled if it is a subset of the positive support of a P-invariant and the weighted token sum in the support of the invariant at an initial marking is greater than zero. An invariant-controlled siphon can never be unmarked at any reachable marking from the initial marking [40], [124], [133], [134]. However, the weight of an arc in a generalized Petri net can be an arbitrarily given positive integer such that it is difficult to properly decide the lower bound of the number of tokens in a siphon.

Barkaoui and Pradat-Peyre are the pioneers who investigate the explicit siphon control problem in generalized Petri nets [10]. A siphon is called max marked at a marking if the number of tokens in its a place is not less than the maximal weight of the arcs from the place. It is said to be max controlled if it is max marked at any reachable marking. A max-controlled siphon can at least fire once a transition in its postset. A Petri net is deadlock free if all siphons are max controlled, i.e., satisfying max cs-property. Similar to the case of ordinary nets, a siphon's controllability condition with respect to a P-invariant is developed. However, such a condition is sufficient but not necessary, and overconservative when it is used to handle deadlock problems in some subclasses of manufacturing-oriented nets [20].

For siphons in S^{4}R, Chao [20] relaxes the max-controllability condition by introducing max$'$-controlled siphons, which is motivated by the concept of deadly marked siphons that can be found by solving an MIP problem [191]. A method to detect a minimal deadly marked siphon is formulated in [269]. Zhong and Li [270] refine the concept of max$'$-controlled siphons and propose a formal description for them. They point out that the marking of the considered resource places satisfying $M(p) \geq \max_{t\in p^\bullet \cap \bar{S}^\bullet}\{W(p,t)\}$ can guarantee that the siphon is sufficiently marked, where $\bar{S}$ is the complementary set of a siphon S. The complementary set of a siphon in a manufacturing-oriented Petri net subclass is defined as a set of activity places that do not belong to the siphon but compete for the resources with the activity places in the siphon. However, this controllability condition of siphons is still restrictive and conservative. Liu *et al*. [173] further relax it by introducing a new condition that is called max$''$ controllability for S^{4}R. A siphon that is max controlled means that it is max$'$ controlled and a max$'$-controlled siphon is accordingly max$''$ controlled. However, the converse is not true. The study in [173] concludes that an S^{4}R net is live if each siphon is max$''$ controlled. An MIP-based method to detect a minimal insufficiently marked siphon is reported in [174], which can be used to find the livelocks in an S^{4}R.

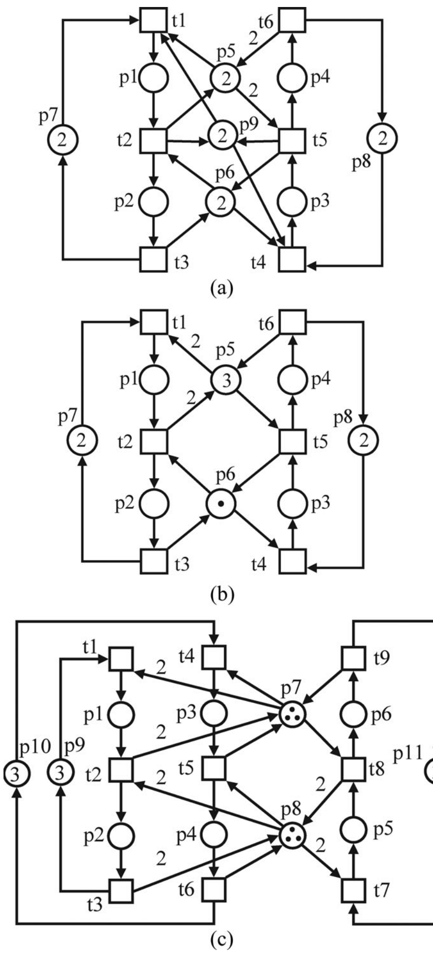

Fig. 4. (a) $S = \{p_2, p_4, p_5, p_6\}$ is max controlled due to place p_9, (b) $S = \{p_2, p_4, p_5, p_6\}$ is max$'$ controlled, and (c) $S = \{p_2, p_4, p_6, p_7, p_8\}$ is max$''$ controlled.

In the net shown in Fig. 4(a), its unique strict minimal siphon is $S = \{p_2, p_4, p_5, p_6\}$. It is max controlled since at any reachable marking M, there exists a place p in S such that $M(p)$ is not less than the maximal weight of the arcs from p. In Fig. 4(b), $S = \{p_2, p_4, p_5, p_6\}$ is a minimal siphon. After a feasible transition sequence $\sigma = t_1 t_4$ fires, a marking $p_1 + p_3 + p_5 + p_7 + p_8$ is reached, at which the siphon is not max marked. Hence, it is not max controlled. However, it is max$'$ controlled and the net is live. In Fig. 4(c), $\{p_2, p_4, p_6, p_7, p_8\}$ is a minimal siphon that does not satisfy the definition of max$'$ controllability. However, it is max$''$ controlled and the net is live.

According to siphon controllability results in a generalized Petri net, Abdallah and ElMaraghy propose a hybrid deadlock-control policy combining prevention and avoidance techniques for S^{4}R [3]. The proposed policy is feasible although there exist some minor technical issues that are reported by Chao [24].

Siphons are well recognized to be tied with deadlocks, which is true in either ordinary or generalized Petri nets. The fact is adequately reported by Reveliotis [205]–[207]. Actually, the siphon control problem in a generalized Petri net is not well addressed yet and further efforts are necessary. Iterative deadlock control is a classic strategy in deadlock prevention. The original work is done by Lautenbach and Ridder [134], which targets bounded and consistent marked Petri nets. If there exists an arc from a place to a transition, whose weight is greater than one, the corresponding transition is split by inserting new places and transitions such that the resulting net is ordinary. Then, all uncontrolled siphons are computed and controlled by adding monitors. Repeat the aforementioned steps until all siphons are controlled. Finally, the split transitions are merged. Such an iterative deadlock-control policy is claimed to be maximally permissive. Note that redundant monitors are common in a controlled system that results from an iterative deadlock-control approach, and the net decomposition approach that is presented in [134] is slightly different from the one in [110] with respect to the introduction of new places and transitions.

Tricas utilizes an iteration method to prevent deadlocks for FMS [222]–[225]. At each iteration step, a siphon is computed and controlled by a monitor, where no net decomposition from a generalized Petri net into a PT-ordinary one is employed. 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 readily comprehensible proof. However, the convergence is not difficult to imagine if the following statements are true:

1) a plant net is bounded;
2) the initial marking is contained in a maximal strongly connected component in its reachability graph;
3) there exists a subgraph of the maximal strongly connected component containing the initial marking such that a Petri net representation can be found for the subgraph that is considered to be a finite automaton; and
4) the unsafe markings are gradually removed as the iterations proceed.

The major disadvantage is that 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 [10], max$'$-controlled [20], or max$''$-controlled siphons [173].

In an iterative deadlock-control algorithm [110], [134], [196], the following operations are usually involved: 1) net transformation, i.e., a generalized net is transformed into an ordinary or PT-ordinary net by adding derivative places and transitions; 2) siphon computation; 3) siphon control by monitors; 4) removal of redundant monitors; and 5) back transformation, i.e., a net is transformed by removing derivative places and transitions.

A recent study finds that if not properly designed, an iterative deadlock control method by using the net decomposition technique in [134] cannot terminate in some cases. The tiny net is shown in Fig. 5. The following typical iterative deadlock-control algorithm does not terminate for the net.

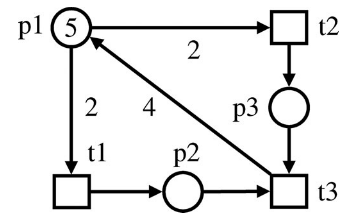

Fig. 5. Generalized net.

Algorithm: Iterative Deadlock Control

while if a net system has reachable dead markings **do**
perform net transformation if it is generalized
find and control siphons by monitors
do back transformation
remove redundant monitors
end while

Motivated by the work proposed in [134], [195], and [196], an iterative siphon control approach to deadlock prevention for a manufacturing-oriented Petri net model by Zhang and Wu [268] can lead to a maximally permissive Petri net supervisor by designing monitors to prevent siphons from being unmarked. At each iteration step, a siphon that is to be controlled is carefully selected by considering the number of tokens in its complementary set. Case studies show that such consideration can usually generate a structurally compact supervisor, i.e., the number of monitors is small. Note that the approach in [268] avoids the complete or partial marking enumeration of a plant model.

In [170] and [171], a polynomial-time deadlock-prevention policy is proposed for a class of FMS with buffers. The idea is to partition those shared buffers for storing different types of parts, thereby destroying the circular-wait condition. However, this policy is not optimal. The research in [172] defines the key-resource/activity-place pairs of S^3PR and proposes a deadlock-prevention policy that is based on this concept. The idea is that we always leave a part requiring key resources for some fixed manufacturing processes, thereby avoiding circular waits. This policy can guarantee that the number of monitors is not greater than the number of resource places. However, it is not optimal and requires that the capacity of each key resource place is bigger than one.

Other active researchers in iterative deadlock prevention are Iordache and Antsaklis [110], [111], [113]. The feature of their research is that no structural constraints are enforced to a Petri net, i.e., their work is applicable to any net. Furthermore, uncontrollable and unobservable transitions are considered. As usual, an iterative approach consists of siphon calculation and control. The major problem is computational complexity, since a complete or partial siphon enumeration is necessary along with incidental structural complexity. Their work notices that there exist redundant monitors in the supervisors when an iterative process terminates. However, the monitor's removal condition is overly strict.

Craig and Zuberek [46] propose an efficient siphon-based deadlock-detection method. They introduce the concept of equivalent siphons and develop two types of net transformation that can reduce equivalent siphons, making the siphon-based deadlock detection much more attractive from a practical point of view.

In addition to the MIP-based deadlock-detection approach that is shown to be, although it is NP-hard in theory, computationally competitive via numerical examples, another significant yet computationally efficient breakthrough in designing liveness-enforcing Petri net supervisors is a deadlock-prevention algorithm with polynomial complexity. Proposed by Park and Reveliotis [189], [191], the thoughts of this algorithm were originally derived from the seminal work in [203] and [204] in an automaton formalism. In other words, the work in [191] can be considered as a Petri net implementation of the deadlock-avoidance algorithm in [203]. The deadlock-control specifications are represented by a set of constraints. In a constraint, the number of tokens in a set of activity places is limited to be a constant. In fact, the deadlock-prevention requirements and specifications are converted into a set of Generalized Mutual Exclusion Constraints (GMEC) that can represent the concurrent use of a finite number of resources shared among different processes [77], [79]. Each GMEC is implemented by a monitor. The number of monitors in the supervisor is equal to that of the resources in the system under consideration. As a result, another advantage of this approach is that the structural-complexity problem of a supervisor is well addressed. However, it causes the behavioral permissiveness issue. Experimental studies show that it is even less permissive than the policy in [59]. Moreover, resource ordering needs to be assigned before the computation of a supervisor. Different resource orderings lead to supervisors with different permissive behavior. In this sense, selecting an optimal resource ordering is of importance to behavioral permissiveness.

The study in [166] proposes the concept of dominated transitions. It indicates that the output arcs of a monitor are not necessarily led to the source transitions of a plant net model as done in [59], which is the reason behind the loss of many permissive states. Dominated transitions are useful to develop a more permissive liveness-enforcing supervisor than the policies in which monitors are led to the source transitions.

Giua and Seatzu discuss the liveness enforcement problem in railway networks by using Petri nets [84]. A supervisor is found by adding appropriate monitors that are designed through siphon analysis. It does not need an exhaustive computation of all siphons. Furthermore, uncontrollable and unobservable transitions in a plant model are considered. The work in [73] considers monitor design for colored Petri nets with applications to deadlock problems for railway networks.

Another interesting field is Internet-motivated video streaming systems, where deadlocks or blocking are caused by network resources with a high-sharing degree [99]. The contributions in [99] are twofold. First a novel class of Petri nets, called non-sequential systems of simple systems with shared resources, and their liveness analysis methods are proposed. Second, a deadlock-prevention policy that is based on generalized elementary siphons is developed. Deadlock problems with production ratio in a system are considered in [104] and [105].

C. Reachability Graph-Based Approaches

As stated previously, behavioral permissiveness is one of the most important criteria in evaluating a supervisor. Determining how to design a maximally permissive, i.e., optimal, Petri net supervisor has been an interesting yet significant problem, from both practical and theoretical points of view. On the one hand, the problem is well addressed in the framework of formal languages and automata in R-W theory [241]. On the other hand, it has remained open in Petri net formalisms for many years. The reasons are twofold. First, Petri nets have limited modeling power, compared with automata. Specifically, it is not true that any automaton can find its free-labeled Petri net implementation. Second, Petri netters do not find effective methods to deal with this problem except some special net structures at particular initial markings [161], [259].

The work by Uzam [227] proposes an approach to design optimal Petri net supervisors by using the theory of regions [5] that originally aims to provide a formal methodology to synthesize a pure Petri net from a transition system. It establishes a connection between transition systems and Petri nets through net synthesis. The idea behind the theory of regions is that a state-based model, a model that describes which states a process can be in and which transitions are possible between these states, can be transformed into a Petri net, a compact representation of the state space, explicitly showing causality, concurrency, and conflicts between transitions. Since its appearance, the theory of regions finds its wide and extensive applications [47]. Shortly after the work in [227], by using popular and plain linear algebra, Ghaffari *et al*. present an easily understandable explanation of the design approach to an optimal liveness-enforcing Petri net supervisor that is based on the theory of regions [76]. An improved version of the work in [227] can be found in [228].

Based on the theory of regions, a design approach of optimal liveness-enforcing Petri net supervisors can be expounded as follows. First, one generates the reachability graph of a plant net model and then finds all marking/transition separation instances (MTSI), as well as the sets of legal and illegal markings. An MTSI takes the form of (M, t), where M is a marking, t is a transition, and $M[t\rangle$ is a forbidden marking that is illegal from the deadlock-control point of view. For an MTSI (M, t), a monitor is computed by solving a linear programming problem (LPP) such that its addition to the plant model disables t at M while ensures the reachability of all legal markings. The fatal disadvantage of the approaches based on the theory of regions is that a complete state enumeration is necessary. As known, the size of the reachability graph of a Petri net grows exponentially with respect to the number of its nodes and initial marking. This is the so-called state explosion problem.

To find an MTSI can be done in polynomial or even linear time by a depth or breadth first search algorithm after a reachability graph is computed. However, for deadlock-control purpose of a Petri net, the number of MTSI is in theory exponential with respect to the size of the model and its initial marking. Hence, the number of LPP that is to be solved is in theory exponential with respect to the plant net size. In this sense, polynomial solvability of an LPP seems meaningless. Moreover, in such an

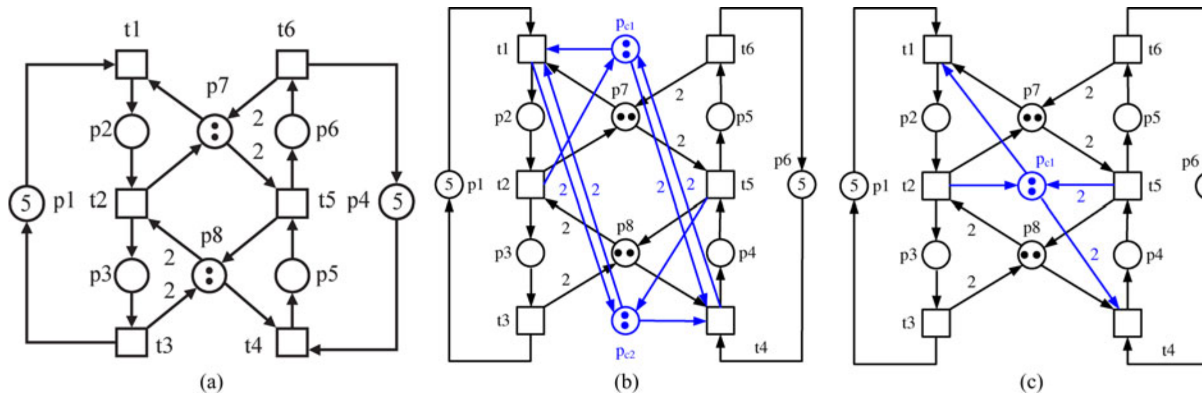

Fig. 6. (a) Petri net without an optimal pure net supervisor. (b) Optimal nonpure supervisor. (c) Most permissive pure net supervisor.

LPP, the number of constraints is almost equal to that of the markings in a state space. Note that the number of LPP to be solved in theory equals to that of MTSI. However, a monitor can implement multiple MTSI, leading to the fact that the number of monitors in a supervisor is generally much smaller than that of MTSI in a reachability graph.

It is worthy to note that even before the publication of the work by Uzam and Ghaffari, a common sense is that any approach that depends on a complete state enumeration is infeasible to real-world production systems.

The approaches in [76] and [227] can find an optimal supervisor if such a supervisor exists. When an optimal supervisor that is expressed by a set of monitors does not exist, the work in [76] and [227] does not offer a deadlock-control solution. For example, one cannot find an optimal Petri net supervisor for the net model shown in Fig. 6(a) by using the theory of regions. In this case, a natural and interesting problem is to find a *most* permissive liveness-enforcing Petri net supervisor such that there are no others that are more permissive than it.

On condition that each monitor is associated with a P-semiflow whose support contains the monitor and activity places in the considered plant that is a manufacturing-oriented Petri net, the work in [35] presents a deadlock-prevention approach to find a maximally permissive liveness-enforcing supervisor for an FMS if such a supervisor exists. Otherwise, it can derive a most permissive one in the sense that there do not exist others that are more permissive than it. Note that a supervisor here is referred to as a pure net.

The proposed approach [35] only computes the reachability graph of a plant model once. By using a vector-covering approach, the sets of legal markings and first-met bad markings (FBM) [231] are reduced to be two further smaller sets, namely, the minimal covering set of legal markings and the minimal covered set of FBM, respectively. At each iteration, an FBM from the minimal covered set is selected. By solving an integer linear programming problem (ILPP), a place invariant (PI) is designed to prevent the FBM from being reachable and no marking in the minimal covering set of legal markings is forbidden. If the ILPP has no solution, implying that there is no maximally permissive Petri net supervisor for the model, another ILPP is designed to remove the least number of legal markings whose reachability conditions contradict others. Then, a PI is redesigned to keep the rest legal markings being reachable. This process is carried out until no FBM can be reached. Finally, a most permissive liveness-enforcing supervisor is obtained.

The net model shown in Fig. 6(a) is a generalized Petri net that has 15 reachable states, 11 of which are safe from the deadlock-control point of view. The methods in [76] and [227] cannot provide a maximally permissive, i.e., optimal, monitor solution for deadlock prevention to the example. However, there does exist an optimal Petri net supervisor that is non-pure, as shown in Fig. 6(b). A most permissive pure net supervisor is shown in Fig. 6(c), where the monitor is associated with a P-semiflow. Such a most permissive supervisor can lead to the fact that the resulting controlled system has nine reachable states. That is to say, there is no other pure net supervisor that is more permissive than it, in which each monitor is associated with a P-semiflow.

Based on the theory of regions, Uzam and Zhou develop an iterative deadlock-prevention approach [230]. It is assumed that there is a monitor solution to the deadlock prevention for a plant net model. The reachability graph of a plant is divided into two parts: deadlock-free zone and deadlock zone. The former is in fact the maximal strongly connected component that contains the initial marking. The latter contains markings from which the initial marking is unreachable. An FBM is defined in the deadlock zone. It does not satisfy deadlock-free control requirements, and its father node is in the deadlock-free zone. Then, deadlocks can be eliminated by preventing the firing of the transitions enabled at an FBM's father marking such that the FBM is unreachable. It is shown that the transition firing control can be converted into a GMEC problem that can be implemented by a monitor [263], [264] whose computation is highly efficient. GMEC are an important specification in supervisory control of discrete-event systems. Other forms of control requirements can be converted into GMEC [114]. Timed GMEC problems are considered in [103].

The approach in [230] suffers from two problems. First, an optimal supervisor cannot be found in general even if such an optimal one exists. Second, at each iteration step, one needs to compute the reachability graph once. It is claimed by the authors that the approach in [230] is efficient. However, its efficiency is naturally questioned since at each iteration step, the computation of a full reachability graph is necessary to verify whether the markings in the deadlock zone are reachable. Note that the work in [227] and [76] needs to compute a reachability graph only once and then to solve an LPP. The work in [229] aims to improve the deadlock-prevention policy by reducing the plant net model as a first step before designing its optimal supervisor.

It is worthy to note that a reachability graph is a reliable, accurate, and effective (surely not efficient) analysis method of a Petri net, although it is computationally expensive. In recent years, abundant contributions have been seen on efficient computation and compact representation of a reachability graph, e.g., the state-space representation based on binary decision diagrams [15], [41], [180], [192], data decision diagrams [44], hierarchical set decision diagrams [45], [88], stubborn sets [232], [233] and sleep set methods for reduced state-space generation [234].

Chen *et al*. [34] present a novel and computationally efficient method to design optimal control places and an iterative approach that computes the reachability graph once only to obtain an optimal supervisor of FMS, which is expressed by a set of monitors. By using a vector-covering approach, the minimal sets of legal markings and FBM are computed. At each iteration, an FBM from the minimal set of FBM is selected. By solving an ILPP, a PI is designed to prevent the FBM from being reached and no marking in the minimal set of legal markings is forbidden. This process is carried out until no FBM can be reached. In order to make the considered problem computationally tractable, BDD is used to compute the sets of legal markings and FBM, and solve the vector-covering problem to obtain the minimal sets of FBM and legal markings. It is worthy to note that this study offers an optimal supervisor for a large Petri net model with 48 places and 38 transitions from [149] by computing all reachable markings.

The deadlock-control policy in [34] is improved in its computational efficiency. By solving an ILPP, a monitor is computed such that as many FBM as possible are prohibited, and no markings in the minimal covering set of legal markings are forbidden, where the monitor is associated with a P-invariant [38]. The objective function of the ILPP maximizes the number of FBMs that are forbidden by the P-invariant. Although the improved method cannot, in general, guarantee the minimality of the supervisory structure, they reduce the overall computational time greatly, which is shown by numerical studies. For a typical FMS example with 26 places and 20 transitions, the time for monitor solution is reduced from over 48 h to a couple of seconds.

The work in [239] can be considered as an improvement of the theory of regions. It first designs a supervisor for a plant net model by using the theory of regions to find maximally permissive behavior. Then, the strict minimal siphons 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 with existing ones in which the theory of regions is used alone, and the permissive behavior of the supervisor is near optimal.

Piroddi *et al*. [195] believe that there are several important drawbacks in the deadlock-prevention methods that are based on elementary siphons, a novel concept in net theory, that are originally developed by Li and Zhou [148], [158], [160], [161]. First, 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. Second, the policies that are based on elementary siphons are usually not maximally permissive. Third, the set of elementary siphons in a Petri net is not unique. Deciding how to select a set of elementary siphons is important, since the selection determines the sets of strongly and weakly dependent siphons. The existence of different sets of elementary siphons also implies that the deadlock-prevention solution is not unique. Last but not the least, the policies that are based on elementary siphons can be applied to some special classes of Petri nets only. Piroddi *et al*. think that it is necessary and reasonable to combine the structural information that is related to strict minimal siphons and reachability analysis that represents the dynamical evolution information in order to reduce the number of iterations in siphon control and the number of monitors.

The work in [195] 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 idea to some extent is similar to that of elementary siphons, although their development is not motivated by them. However, it considers the relationship between uncontrolled siphons and deadlock markings such that it becomes an *accurate* deadlock control method. On the one hand, it needs less monitors than the existing methods do. On the other hand, the resulting supervisor is highly permissive. Although it is not formally proved to be optimal, it is at least near optimal. It is worthy to note that the work provides an optimal supervisor for a well-known FMS example originally presented in [59], which is computed within a reasonable time. Before this work, the most permissive but not optimal supervisor for this example is offered by Uzam and Zhou in [229] and [230].

The major technical problem in [195] is 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. In the computational overhead aspect, the comparison between the policy based on the theory of regions and this one remains open. After the complexity problem is recognized, Piroddi *et al*. shortly improve the method [196] by using the MIP-based deadlock detection approach such that the complete minimal siphon enumeration is avoided. From the case study, the improved version of the combined siphon and marking policy is computationally competitive.

Recently, Hu and Li [100] have studied the deadlock-prevention problem for a class of conjunctive/disjunctive resource allocation systems in which multiple resource acquisitions and flexible routes are allowed, representing a large class of FMS. Local and global control approaches can be derived. Compared with those reported in the literature, the proposed methods can well balance among computational complexity, behavioral permissiveness, and structural complexity. The local one can lead to a nearly optimal Petri net supervisor with reasonable computational efficiency. The global one can result in a less permissive supervisor but with simple structural configuration. Moreover, both local and global supervisors involve only limited online computation cost. Similar work that avoids a complete siphon or state enumeration can be found in [101].

D. Elementary Siphon-Based Approach

Even in a Petri net with a rather simple structure such as linear S^3PR, the number of its strict minimal siphons in the worst case is proved to be exponential with respect to its size [237]. If all the strict minimal siphons are explicitly controlled without considering any difference or relationship among them, the resulting supervisor is structurally complex in theory, as shown in [59]. For example, an automated manufacturing system with 16 machine tools and 13 robots is considered. Its Petri net model consists of 128 places and 88 transitions. According to the deadlock-prevention policy in [59], the supervisor contains 587 monitors and 15 464 arcs. One can imagine its huge structural complexity.

To alleviate such a problem, Li and Zhou propose the concept of elementary siphons [148], [151], [161]. For a deadlock-control purpose, problematic siphons (that can cause dead transitions) in a Petri net are divided into elementary and dependent ones. The latter is originally named as *redundant* siphons [148]. By the incidence matrix of a Petri net, Li and Zhou define the characteristic T-vector of a siphon, which is the sum of the rows corresponding to the places in the siphon, and indicates the change of the number of tokens in it when a transition fires. The characteristic T-vectors of all problematic siphons constitute a matrix called their characteristic T-vector matrix. From the matrix, a maximal linearly independent set of vectors can be found. The siphons corresponding to the vectors in this set are said to be elementary. The others are called dependent siphons that are further distinguished into weakly and strongly dependent ones by deciding whether the linear combination coefficients are all positive or not.

Two key contributions that underlie the concept of elementary siphons are as follows: 1) The number of elementary siphons in a Petri net is bounded by the smaller of the place and transition counts; and 2) a dependent siphon can be implicitly controlled by controlling its elementary ones. For deadlock prevention that is achieved by monitors, it is of significance that a dependent siphon can be controlled via explicit control of its elementary siphons by designing monitors and properly setting their control depth variables. An FMS example is investigated in [148] with its Petri net model having 26 places, 20 transitions, and 18 strict minimal siphons. By using the concept of elementary siphons, a liveness-enforcing Petri net supervisor is computed by explicitly adding only six monitors to control the six elementary siphons among 18 strict minimal siphons. However, the work in [59] needs to design monitors for all 18 strict minimal siphons.

In [148], the controllability of a dependent siphon is explored with respect to elementary siphons that are invariant controlled. A more general controllability condition than that in [148] is developed in [160] for dependent siphons in ordinary Petri nets. In [158], Li and Zhao extend such results to generalized nets, which is based on the max-controlled siphons [10]. However, the computational complexity of supervisor design in [160] remains to be exponential with respect to the net size, since the computation of a set of elementary siphons depends on complete siphon enumeration.

The computational efforts for a supervisor based on elementary siphons are reduced by introducing a siphon solution technique using MIP-based deadlock detection. The study in [153] proposes an iterative deadlock-prevention policy by using the concept of elementary siphons. At each iteration, a maximal unmarked siphon is found by solving an MIP problem. Then, a strict minimal siphon is extracted from the maximal unmarked one. If the siphon extracted is elementary with respect to the computed ones, it is explicitly controlled by a monitor. If it is dependent, its controllability is decided by checking whether it needs to be explicitly controlled. The iteration process terminates when no unmarked siphon is found in the controlled Petri net with monitors.

The work in [153] to a large extent reduces the computational cost to design a supervisor and the resulting supervisor's structural complexity compared with [59]. However, it does not improve the behavioral permissiveness. The work in [150] develops a two-phase deadlock-prevention policy. The first phase adds a monitor for each elementary siphon that is derived from the MIP-based deadlock-detection method. The output arcs of a monitor are led to the source transitions of a plant net model, which represent the entry of raw parts into a system. The second phase rearranges the output arcs of the monitors such that the transitions with which they are associated are away from the source transitions as far as possible if this rearrangement does not result in dead transitions. Such an improvement increases the behavioral permissiveness of a supervisor. The policies that underlie the idea of elementary siphons can also be found in [152] and [266].

Hu and Li [102] extend the study in [153] to a more general class of Petri nets, i.e., S^4PR. Insufficiently marked siphons are used to characterize deadlocks, which can be found by solving an MIP problem. A generalized net is first transformed into an ordinary one before the MIP problem proceeds.

The work in [163] can be considered as an application of the divide-and-conquer strategy in the deadlock-control area for resource allocation systems, which is an important problem solution paradigm in computer science. A plant net model is divided into an idle subnet, an autonomous subnet, and a number of small but independent subnets, which are called toparchies, by the concept of resource circuits. A liveness-enforcing supervisor, which is called a toparch, is designed for each toparchy. If a particular separation condition holds in a plant net model, the computational overhead of toparches is significantly reduced. This research shows that the resulting net by composing the toparches that are derived for the toparchies can serve as a liveness-enforcing Petri net supervisor for the whole plant model. A case study shows the significance of the divide-and-conquer strategy via a number of typical deadlock-prevention policies.

It is not surprising that there exist redundant monitors in a liveness-enforcing Petri net supervisor, particularly when it is derived from an iterative siphon control approach. In [231], a redundant monitor is identified and eliminated by computing the reachability graph of a controlled system. If the removal of a monitor does not lead to the loss of liveness of the controlled system, it can be removed from the supervisor. The major drawback is the computational-complexity problem, since the reachability graph and liveness check are necessary. In order to avoid a complete state enumeration, Li and Hu [164] propose two methods to remove monitors from a Petri net supervisor. The first is based on the concept of implicit places [74], [201]. It is shown that the implicity of a monitor is decided by solving an LPP that can be done in polynomial time. The second is derived from the MIP-based deadlock-detection method. If the removal of a monitor does not change the optimal solution of an MIP problem that is derived from the controlled system, then it is implicit or its removal may lead to more permissive behavior while liveness is preserved.

As known, dependent siphons can be further divided into strongly and weakly dependent ones. An interesting work is done by Chao and Li in [26], which explores the structural condition in a class of Petri nets under which there exists a set of elementary siphons such that all the others are strongly dependent.

The solution for elementary siphons in a Petri net is significant to the development of deadlock-prevention policies [98]. By using the concept of handles and bridges [57], much work is done by Chao on the computation of minimal and elementary siphons in a resource allocation system [18], [21]–[23], [25], [29], [157]. The algorithm that is proposed by Wang *et al*. [237] is also of polynomial complexity, which can find a set of elementary siphons for linear S^3PR. A significant result in [237] is the supremum of the number of strict minimal siphons in a linear S^3PR, which is $2^n - n - 1$, where n is the number of resource places.

A known result on deadlock freedom in net theory is Commoner's theorem, i.e., an ordinary net is deadlock free if there are no siphons that can be unmarked. For many subclasses of Petri nets, the fact that there is no unmarked siphon implies their liveness. The subclasses include PPN [7], S^3PR [59], ES^3PR [107], and ERCN-merged nets [258]. The work in [19] proposes the concept of virtual first-order structures. A net without virtual first-order structures has the property that the absence of unmarked siphons at any reachable marking implies the liveness. Furthermore, it is improved in recent work [28] by showing that a nonvirtual net is live as long as all its siphons can never be emptied, which generalizes the existing net subclasses that has such a property. Other deadlock-prevention policies that are developed by Chao can be seen in [30] and [31].

The concept of elementary siphons aims to simplify the structure of a liveness-enforcing Petri net supervisor, which can also be used in deadlock prevention for timed Petri nets [86]. The number of transitions in a supervisor is not greater than that of a plant model. As a result, its structural complexity is usually evaluated by the number of its monitors. However, in a general case, a deadlock-prevention policy that is based on elementary siphons cannot find a minimal supervisory structure. The work by Chen and Li in [36], by solving MIP problems, finds an optimal liveness-enforcing supervisor that has a minimal number of monitors under the condition that each monitor is associated with a P-semiflow whose support contains the monitor and some activity places. For the well-known FMS example whose Petri net model has 26 places and 20 transitions [59], an optimal supervisor with a minimal number of monitors, i.e., five, can be found. However, an open problem is whether there exists an optimal supervisor with a more compact supervisory structure, in which a monitor is not associated with a PI or a P-semiflow. Another interesting problem is whether there exists an optimal supervisor in which the number of monitors is definitely bounded by the size of a plant. The design of structurally simple supervisors for a set of GMEC problems can be found in [167].

E. Combined Techniques

To reduce the number of MTSI when using the theory of regions, Li *et al*. develop a two-phase deadlock-prevention policy by siphon control and the theory of regions [159]. First, strict minimal siphons are identified through resource circuits only and controlled by monitors, whose quantity is bounded by the smaller of place and transition counts. This leads to the fact that siphon identification and control is of polynomial complexity. Then, the theory of regions is applied to the augmented Petri net with monitors to find a supervisor. Since the siphon control in the first phase is optimal from deadlock-prevention point of view, the final supervisor is optimal if such a supervisor exists.

Motivated by the tight connections between directed graphs (digraphs) and Petri nets in deadlock control for FMS [68], Maione and DiCesare [177] propose a hybrid deadlock-prevention policy by using directed graphs and Petri nets, taking advantages of the strong points of both techniques. In order to avoid searching siphons in the Petri net model of a system, deadlock detection is carried out by analyzing the structures of the digraph that models the system. Then, the digraph-based information is translated into the deadlock marking of the corresponding Petri net, which is used to design monitors to prevent empty siphons. Finally, a number of new control policies are developed, which are less restrictive than other efficient policies [177].

The work in [169] presents a novel deadlock-detection approach for WS^3PR, which is an extension of S^3PR with weighted arcs. It explores the numerical relationships among weights, and between weights and initial markings based on simple circuits of resource places, which are the simplest structure of a circular wait, rather than siphons. It is shown that a WS^3PR is deadlock free and live if it satisfies a proposed condition with respect to arc weights and initial markings. A set of polynomial algorithms are developed to implement the proposed method.

VI. Open Problems

A. Uncontrollable and Unobservable Transitions

Uncontrollable and unobservable events in a plant may be present. Accordingly, it is reasonable and practical to consider their existence in a Petri net model of an FMS. Note that in R-W theory [199], [241], uncontrollable and unobservable events are sufficiently considered. However, Petri net researchers usually assume that all transitions are controllable and observable when a deadlock-prevention policy is developed for an FMS. When the presence of uncontrollable and unobservable transitions is taken into account, most existing deadlock-control policies need to be refined or even reinvestigated. For example, an optimal supervisor cannot be found if some transitions are uncontrollable in a Petri net model [165].

The work in [110] considers the design of a liveness-enforcing supervisor for a Petri net with uncontrollable and unobservable transitions. A siphon's controllability is represented by a linear constraint on markings. Two conditions are proposed to decide the admissibility of a linear constraint. An admissible constraint is enforced directly, and an inadmissible one can be performed after it is transformed to be admissible by using the constraint transformation method in [182] and [183]. However, the deadlock-prevention policy in [110] is performed in an iterative way. Also, it suffers from a computational-complexity problem since at each iteration step, a complete siphon enumeration is needed.

The work in [198] considers the applicability of a deadlock-prevention policy that is developed under the assumption that all transitions are controllable and observable, to a plant with uncontrollable and unobservable transitions. The concept of critical controllable and observable transitions is proposed. It is concluded that a policy that is developed without considering uncontrollable and unobservable transitions is applicable to a plant with uncontrollable and unobservable transitions if and only if those in the set of critical controllable transitions are controllable and those in the set of critical observable transitions are observable.

B. Fault-Tolerant Deadlock Control

The selection of deadlock-control strategies depends on the frequency of deadlock occurrences in a system. If deadlocks are rather rare, a time-out mechanism may be accepted as the best approach to deal with deadlocks due to its low overhead. This is deadlock detection and recovery. In some cases, such a strategy is not permitted due to technical or other factors. Instead, deadlocks are expected to forbid even if some resources break down. In a contemporary manufacturing system, automated equipment is widely and extensively used. The occurrences of faults in unreliable devices and machines can falsify a correctly designed deadlock-prevention policy [143]. Robust deadlock-prevention and avoidance policies that consider various errors and faults in an FMS are an interesting topic by using Petri nets as a formalism [96].

C. Existence of Optimal Supervisors

The existence of marking-based, not monitor-based, liveness-enforcing supervisors for discrete-event systems modeled with Petri nets is investigated by Sreenivas [216]–[218] in which the computation of a reachability graph is necessary. The problem is also discussed by Giua and DiCesare from the formal language point of view [78]. Iordache and Antsaklis develop generalized conditions for liveness enforcement and deadlock prevention in Petri nets [109], which are based on the concept of active subnets and siphons. However, no sufficient attention is paid to the existence of an optimal liveness-enforcing Petri net supervisor for FMS, which is expressed by a set of monitors. A natural and interesting problem is the structural and initial marking conditions of a Petri net under which there exists such a supervisor. For example, whether there is an optimal supervisor for any S^3PR with an acceptably initial marking is interesting. For the existing manufacturing-oriented ordinary Petri net subclasses in the literature such as PPN [7] and S^3PR [59], we have not seen any example whose optimal ones do not exist. However, it is easy to find an S^4PR whose optimal supervisors that are represented by pure Petri nets do not exist. The net shown in Fig. 6(a) is such an example. Of course, there exists an optimal one that takes the form of an automaton. In fact, such an automaton cannot admit a free-labeled pure Petri net representation.

If the reachability graph of a Petri net has a maximal strongly connected component that contains the initial marking and all transitions are controllable, an optimal marking-based supervisor, i.e., an optimal supervisor that takes the form of an automaton, exists. An interesting issue is the relationship between monitors and marking-based supervisors-enforcing liveness. For a bounded ordinary Petri net, it is shown in [89] that 1) there exists a liveness-enforcing monitor if and only if there exists an optimal marking-based liveness-enforcing supervisor; and 2) a liveness-enforcing monitor may not be optimal. The results in [89] are established by net unfolding techniques [178], [179] that map a Petri net to an acyclic occurrence net. A finite prefix of the occurrence net is defined to give a compact representation of the Petri net's reachability graph while preserving the causality between net transitions. This approach is used to deal with deadlock problems in [82], [90], [91], and [219]. A number of problems remain open. For example, it is appealing to find the condition under which an optimal monitor-based supervisor can be computed once an optimal marking-based supervisor exists.

D. Deadlock Prevention Under Dynamic Control Specifications

In supervisory control of discrete-event systems, an important and typical class of control specifications is linear inequalities on markings of a Petri net [114], which is the so-called GMEC [78]. Other forms of control constraints can be transformed into GEMC problems [112]. The control specifications are assumed to be stable during the design and runtime phases of a supervisor. However, in practice, control requirements can change during the runtime phase of a supervisor. For example, they can vary because of breakdown of a machine tool or client order change. As far as the authors know, no attention is paid to the deadlock-control problem in FMS under dynamic control specifications.

VII. Concluding Remarks

Absence of deadlocks is critical in systems that are expected to operate in an automated way. They include life-support systems, nuclear plants, transportation control systems, and automated manufacturing systems. A systematic and efficient method to prevent, avoid, and detect deadlocks is of primary importance for them.

With the wide application of agile and flexible production modes, ignoring deadlocks in FMS is usually not a preferable option. Over the past two decades, deadlock-control research received much attention from academic and industrial communities, leading to ample resolution methodologies, most of which were based on Petri nets. This study aims to present a literature review of deadlock-control strategies for FMS with a focus on deadlock prevention.

As a structural object, siphons are tied to dead transitions whose existence leads to the loss of liveness of a Petri net. Siphon control provides an effective way to prevent the occurrence of deadlocks. In a general case, a siphon-based deadlock-prevention policy cannot find an optimal, i.e., maximally permissive, supervisor, particularly, for generalized Petri nets. However, it can represent the liveness requirements as a set of linear inequality constraints with respect to initial markings. For a fixed net structure with an initial marking, once the liveness requirements are established, it is easy to decide its supervisor when the initial marking changes.

It is shown that reachability graph-based approaches can usually find an optimal supervisor if it exists. However, they suffer from expensive overhead, since the complete state enumeration of a Petri net is exponential with its size and initial marking. The deadlock-prevention policies that use partial reachability graphs seems to provide a tradeoff between computational cost and behavioral permissiveness. For a fixed net structure with a new initial marking, all the computation needs to be carried out afresh, since a reachability graph is sensitive to both net structure and initial marking. This is proved to be computationally inferior in comparison with siphon-based strategies, since siphons are a pure structural objects whose computation is independent of initial markings.

Deadlock prevention through a proper configuration of initial markings is overly conservative. A hybrid deadlock-control approach refers to the one that either combines deadlock prevention and avoidance or utilizes two different formal paradigms. The essential motivation of such a policy is to take the advantages of multiple strategies or formalisms, while avoiding their disadvantages. This paper should facilitate engineers in choosing a suitable deadlock control method for their industrial application cases.

VIII. Bibliography Notes

The notable yet original textbooks on Petri nets are [194] and [202] that cover modeling issues and analysis techniques. The monograph [127] focuses on colored Petri nets and their applications. Books or book chapters about Petri nets are [4] and [236] on performance evaluation and modeling, [48] on Petri nets and logic controller design, [274] on Petri net synthesis for manufacturing systems, [275] on Petri nets and agile automation, [276] on modeling, simulation, and control of FMS, [50] on modeling, analysis, and performance evaluation, [161] on deadlock resolution in automated manufacturing systems that are based on Petri nets, [278] on various deadlock problems and their solutions in computer-integrated systems, [94] on Petri nets and other tools as formalisms of discrete-event systems, and [51] and [197] on the applications of Petri nets to various issues in manufacturing. A general introduction book to discrete-event systems can be found in [16], [94], and a book on resource allocation system management from discrete-event system point of view can be found in [207].

The survey papers in journals are [185] and [193] on basics of Petri nets and their analysis techniques, [212] and [215] on Petri nets and production systems, [54] on Petri nets in manufacturing system control, [156] on the comparison of deadlock-prevention policies that are based on Petri nets via a typical case study, [72] on deadlock control methods in automated manufacturing systems, [184] on modeling FMS using Petri nets, [284] on Petri nets and their industrial applications and [92] on Petri nets as a formalism for discrete-event systems.

Many papers in special issues of journals, or special sessions in international conferences or workshops are devoted to Petri nets and manufacturing such as [33], [49], [50], [121], [125], [126], [130], [213], [277], [279], [280], and [282]. Survey papers on Petri nets from a system theory perspective can be found in [83] and [214].

As a structural object, siphons play an important role in the analysis of structural and behavioral properties of Petri nets. An algorithm with polynomial complexity to decide whether a set of places is a minimal siphon can be found in [8]. Classic and typical siphon computation methods are presented in [58], [132], [238], and [265]. A siphon computation method that is claimed to be rather efficient is developed in [43], which can find 2×10^7 siphons within 1 h. For a class of Petri nets, a siphon solution approach is given in [2] that is also efficient through experimental studies. A parallel solution to compute siphons is established by Tricas and Ezpeleta [226]. An efficient minimal siphon computation approach by using BDD is provided in [37]. Resource-transition circuits can be used to characterize deadlocks [260]. The relationship between maximal perfect resource-transition circuits and siphons is explored in [261].

APPENDIX

Basics of Petri Nets

A Petri net [185] is a four-tuple $N = (P, T, F, W)$, where P and T are finite and nonempty sets. P is the set of places, and T is the 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 flow relation of the net, which is 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 called an ordinary net, which is denoted as $N = (P, T, F)$, if $\forall f \in F, W(f) = 1$. A marking M of N is a mapping from P to $\mathbb{N}$. (N, M_0) is called a net system or marked net, which is sometimes denoted by $G = (P, T, F, W, M_0)$. For economy of space, we use $\sum_{p \in P} M(p)p$ to denote vector M.

Let $x \in P \cup T$ be a node of $N = (P, T, F, W)$. The preset of x is defined as ${}^\bullet x = \{y \in P \cup T | (y, x) \in F\}$, and the postset of x is defined as $x^\bullet = \{y \in P \cup T | (x, y) \in F\}$. This notation can be extended to a set of nodes as follows: given $X \subseteq P \cup T$, ${}^\bullet X = \cup_{x \in X} {}^\bullet x$, and $X^\bullet = \cup_{x \in X} x^\bullet$. Note that ${}^{\bullet\bullet} X$ is the preset of ${}^\bullet X$, and $X^{\bullet\bullet}$ is the postset of $X^\bullet$. Given place p, we denote $\max\{W(p, t) \mid t \in p^\bullet\}$ by $\max_{p^\bullet}$.

A transition $t \in T$ is enabled at a marking M if $\forall p \in {}^\bullet t$, $M(p) \geq W(p, t)$. This fact is denoted as $M[t\rangle$. Firing t yields a new marking M' such that $\forall p \in P$, $M'(p) = M(p) - W(p, t) + W(t, p)$, which is denoted as $M[t\rangle M'$. 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 $M_1, M_2, \ldots$, and M_n such that $M[t_0\rangle M_1 [t_1\rangle M_2 \ldots M_n [t_n\rangle M'$ holds. The set of markings reachable from M in N is denoted as $R(N, M)$.

A net is pure (self-loop free) if $\nexists x, y \in P \cup T$, $\ni (x, y) \in F \wedge (y, x) \in F$. A pure net $N = (P, T, F, W)$ can be alternatively represented by its incidence matrix $[N]$, where $[N]$ is a $|P| \times |T|$ integer matrix with $[N](p, t) = W(t, p) - W(p, t)$, where $|P|$ $(|T|)$ means the cardinality of set P (T). For a place p in net N, its incidence vector is denoted as $[N](p, \cdot)$.

A transition $t \in T$ is live at M_0 if $\forall M \in R(N, M_0)$, $\exists M' \in R(N, M)$, $\ni M'[t\rangle$. N is dead at M_0 if $\nexists t \in T$, $\ni M_0[t\rangle$. (N, M_0) is deadlock free if $\forall M \in R(N, M_0), \exists t \in T, \ni M[t\rangle$. (N, M_0) is live if $\forall t \in T$, t is live at M_0. (N, M_0) is bounded if $\exists k \in \mathbb{N}$, $\forall M \in R(N, M_0)$, $\forall p \in P$, $\ni M(p) \leq k$. (N, M_0) is quasi-live if $\forall t \in T, \exists M \in R(N, M_0), M[t\rangle$ holds. (N, M_0) is said to be reversible, if for each marking $M \in R(N, M_0)$, M_0 is reachable from M.

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. A P(T)-vector $I(J)$ is denoted by $\sum_{p \in P} I(p)p$ $(\sum_{t \in T} J(t)t)$ for economy of space. We denote column vectors where every entry equals $0(1)$ by $\mathbf{0}(\mathbf{1})$. I^T and $[N]^T$ are the transposed versions of a vector I and matrix $[N]$, respectively. P-vector I is a P-invariant (place invariant, or PI for short) if $I \neq \mathbf{0}$ and $I^T[N] = \mathbf{0}^T$. P-invariant I is said to be a P-semiflow if no element of I is 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, whereas $\|I\|^- = \{p | I(p) < 0\}$ denotes the negative support of I. An invariant is said to be minimal when its support is not a strict superset of the support of any other, and the greatest common divisor of its elements is one. If I is a P-invariant of (N, M_0), then $\forall M \in R(N, M_0)$, $I^T M = I^T M_0$.

Let $N = (P, T, F, W)$ be a Petri net with $P_X \subseteq P$ and $T_X \subseteq T$. $N|_{P_X \cup T_X} = (P_X, T_X, F_X, W_X)$ is called a subnet generated by $P_X \cup T_X$ if $F_X = F \cap [(P_X \times T_X) \cup (T_X \times P_X)]$, $\forall f \in F_X$, $W_X(f) = W(f)$.

Let S be a nonempty subset of places. $S \subseteq P$ is a siphon (trap) if ${}^\bullet S \subseteq S^\bullet (S^\bullet \subseteq {}^\bullet S)$. A marked trap can never be emptied. A siphon is said to be minimal if it contains no other siphons as its proper subset. A minimal siphon is strict if it contains no trap. A siphon is said to be controlled if it can never be emptied. A siphon S is said to be invariant controlled by P-invariant I if $I^T M_0 > 0$ and $\|I\|^+ \subseteq S$.

Let (N, M_0) be a generalized Petri net and S be a siphon of N. S is max controlled if $\forall M \in R(N, M_0)$, $\exists p \in S$, $M(p) \geq \max_{p^\bullet} = \max_{t \in p^\bullet} W(p, t)$. (N, M_0) is said to satisfy the cs-property if each minimal siphon of N is max controlled. If (N, M_0) is live, then it satisfies the cs-property. If (N, M_0) satisfies the cs-property, then it is deadlock free [10].

ACKNOWLEDGMENT

We would like to extend our sincere gratitude to *Research in Paris* program, Visiting Professor Program launched by King Saud University, Alexander von Humboldt Research Grant, Alexander von Humboldt Foundation, Germany, and Cai Yuan-Pei Program co-launched by China and France. The authors would like to thank Prof. Y. Chao, Prof. W. Wu, Prof. Y. Huang, Prof. K. Xing, Prof. K. Barkaoui, Prof. M. Uzam, Prof. H.-M. Hanisch, Prof. G. Liu, Prof. M. P. Fanti, Prof. S. Wang, Prof. K. Mohamed, Prof. O. Mosbahi, and Prof. Z. Zhang for their constructive criticisms and insights. The authors are also thankful to Prof. J. Shao, Xi'an JiaoTong University, Xi'an, China, for her assistance in literature search. The authors would like to thank the anonymous referees whose comments and suggestions helped us improve the quality and presentation of this paper.

REFERENCES

[1] W. M. P. van der Aalst, *Workflow Management: Models, Methods, and Systems*. Cambridge, MA: MIT Press, 2004.

[2] I. B. Abdallah, H. A. ElMaraghy, and T. ElMekkawy, "A logic programming approach for finding minimal siphons in S^3PR nets applied to manufacturing systems," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Orlando, FL, 1997, pp. 1710–1715.

[3] I. B. Abdallah and H. A. ElMaraghy, "Deadlock prevention and avoidance in FMS: A Petri net based approach," *Int. J. Adv. Manuf. Technol.*, vol. 14, no. 10, pp. 704–715, 1998.

[4] M. A. Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis, *Modelling with Generalized Stochastic Petri Nets*. New York: Wiley, 1995.

[5] E. Badouel and Ph. Darondeau, "Theory of regions," in *Lectures on Petri Nets I: Basic Models*, LNCS 1491, W. Reisig and G. Rozenberg, Eds. Berlin, Germany : Springer, 1998, pp. 529–586.

[6] Z. Banaszak and E. Roszkowska, "Deadlock avoidance in pipeline concurrent processes," *Podstawy Sterowania (Found. Control)*, vol. 18, no. 1, pp. 3–17, 1988.

[7] Z. Banaszak and B. H. Krogh, "Deadlock avoidance in flexible manufacturing systems with concurrently competing process flows," *IEEE Trans. Robot. Autom.*, vol. 6, no. 6, pp. 724–734, Dec. 1990.

[8] K. Barkaoui and B. Lemaire, "An effective characterization of minimal deadlocks and traps in Petri nets based on graph theory," in *Proc. 10th Int. Conf. Appl. Theory Petri Nets*, Bonn, Germany, 1989, pp. 1–21.

[9] K. Barkaoui and I. B. Abdallah, "A deadlock prevention method for a class of FMS," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Vancouver, Canada, 1995, pp. 4119–4124.

[10] K. Barkaoui and J. F. Pradat-Peyre, "On liveness and controlled siphons in Petri nets," in *Proc. 17th Int. Conf. Appl. Theory Petri Nets*, LNCS 1091, Osaka, Japan, 1996, pp. 57–72.

[11] K. Barkaoui, A. Chaoui, and B. Zouari, "Supervisory control of discrete event systems based on structure theory of Petri nets," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Orlando, FL, 1997, pp. 3750–3755.

[12] K. Barkaoui and L. Petrucci, "Structural analysis of workflow nets with shared resources," in *Proc. Workshop Workflow Manage.: Net-Based Concepts, Models, Techn. Tools*, Lisbon, Portugal, 1998, pp. 82–95.

[13] F. Basile, P. Chiacchio, A. Giua, and C. Seatzu, "Deadlock recovery of Petri net models controlled using observers," in *Proc. IEEE Symp. Emerg. Technol. Factory Autom.*, 2001, vol. 2, pp. 441–449.

[14] C. Basnet and J. H. Mize, "Scheduling and control of flexible manufacturing systems: A critical review," *Int. J. Comput. Integr. Manuf.*, vol. 7, no. 6, pp. 340–355, Nov. 1994.

[15] R. Bryant, "Graph-based algorithms for boolean function manipulation," *IEEE Trans. Comput.*, vol. C-35, no. 8, pp. 677–691, Aug. 1986.

[16] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 2nd ed. New York: Springer-Verlag, 2008.

[17] A. Castelnuovo, L. Ferrarini, and L. Piroddi, "An incremental Petri nets approach to the modeling of manufacturing systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 4, no. 3, pp. 424–434, Jul. 2007.

[18] D. Y. Chao, "Computation of elementary siphons for deadlock control," *Comput. J.*, vol. 49, no. 4, pp. 470–479, 2006.

[19] D. Y. Chao, "Maximal class of weakly live nets with nonempty siphons," *IEEE Trans. Syst., Man, Cybern. B*, vol. 36, no. 6, pp. 1332–1341, Dec. 2006.

[20] D. Y. Chao, "Max′-controlled siphons for liveness of S^3PGR^3," *IET Control Theory Appl.*, vol. 1, no. 4, pp. 933–936, 2007.

[21] D. Y. Chao, "A graphic-algebraic computation of elementary siphons of BS^3PR," *J. Inf. Sci. Eng.*, vol. 23, no. 6, pp. 1817–1831, 2007.

[22] D. Y. Chao, "An incremental approach to extract minimal bad siphons," *J. Inf. Sci. Eng.*, vol. 23, no. 1, pp. 203–214, 2007.

[23] D. Y. Chao, "Searching strict minimal siphons for SNC-based resource allocation systems," *J. Inf. Sci. Eng.*, vol. 23, no. 3, pp. 853–867, 2007.

[24] D. Y. Chao, "Comments on 'Deadlock prevention and avoidance in FMS: A Petri net based approach,' " *Int. J. Adv. Manuf. Technol.*, vol. 39, nos. 3–4, pp. 317–318, 2008.

[25] D. Y. Chao, "Incremental approach to computation of elementary siphons for arbitrary S^3PR," *IET Control Theory Appl.*, vol. 2, no. 2, pp. 168–179, 2008.

[26] D. Y. Chao and Z. W. Li, "Structural conditions of systems of simple sequential processes with resources nets without weakly dependent siphons," *IET Control Theory Appl.*, vol. 3, no. 4, pp. 391–403, 2009.

[27] D. Y. Chao, "Reducing MIP iterations for deadlock prevention of flexible manufacturing systems," *Int. J. Adv. Manuf. Technol.*, vol. 41, nos. 3–4, pp. 343–346, 2009.

[28] D. Y. Chao, "Automated manufacturing system: Virtual-nets or non-virtual-nets?" *IET Control Theory Appl.*, vol. 3, no. 6, pp. 671–680, 2009.

[29] D. Y. Chao, "Direct minimal empty siphon computation using MIP," *Int. J. Adv. Manuf. Technol.*, vol. 45, nos. 3–4, pp. 397–405, 2009.

[30] D. Y. Chao, "Weighted characteristic P-vector and deadlock control of WS^3PR," *J. Inf. Sci. Eng.*, vol. 26, no. 3, pp. 1121–1136, 2010.

[31] D. Y. Chao, "Fewer monitors and more efficient controllability for deadlock control in S^3PGR^2 (systems of simple sequential processes with general resource requirements)," *Comput. J.*, vol. 53, pp. 1783–1798, 2010.

[32] J. Charr, D. Teichroew, and R. Volz, "Developing manufacturing control software: A survey and critique," *Int. J. Flexible Manuf. Syst.*, vol. 5, no. 1, pp. 53–88, 1993.

[33] F. Chen, "Petri nets applications in automated manufacturing systems," *Int. J. Adv. Manuf. Technol.*, vol. 14, no. 10, 1998 (special issue).

[34] Y. F. Chen, Z. W. Li, M. Khalgui, and O. Mosbahi, "Design of a maximally permissive liveness-enforcing Petri net supervisor for flexible manufacturing systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 3, pp. 374–393, Apr. 2011.

[35] Y. F. Chen and Z. W. Li, "Most permissive liveness-enforcing Petri net supervisors for flexible manufacturing systems," *Int. J. Prod. Res.*, 2011, to be published.

[36] Y. F. Chen and Z. W. Li, "Design of a maximally permissive liveness-enforcing supervisor with a compressed supervisory structure for flexible manufacturing systems," *Automatica*, vol. 47, no. 5, pp. 1028–1034, 2011.

[37] Y. F. Chen and G. Y. Liu, "Computation of minimal siphons in Petri nets by using binary decision diagrams," *ACM Trans. Embedded Comput. Syst.*, 2011, to be published.

[38] Y. F. Chen, Z. W. Li, and M. C. Zhou, "Behaviorally optimal and structurally simple liveness-enforcing supervisors of flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A: Syst. Humans*, 2011, to be published.

[39] H. Cho, T. K. Kumaran, and R. A. Wysk, "Graph-theoretic deadlock detection and resolution for flexible manufacturing systems," *IEEE Trans. Robot. Autom.*, vol. 11, no. 3, pp. 413–421, Jun. 1995.

[40] F. Chu and X. L. Xie, "Deadlock analysis of Petri nets using siphons and mathematical programming," *IEEE Trans. Robot. Autom.*, vol. 13, no. 6, pp. 793–804, Dec. 1997.

[41] G. Ciardo, "Reachability set generation for Petri nets: Can brute force be smart?" in *Proc. 25th Int. Conf. Appl. Theory Petri Nets*, LNCS 3099, , 2004, pp. 17–34.

[42] E. G. Coffman, M. J. Elphick, and A. Shoshani, "Systems deadlocks," *ACM Comput. Surv.*, vol. 3, no. 2, pp. 66–78, 1971.

[43] R. Cordone, L. Ferrarini, and L. Piroddi, "Enumeration algorithms for minimal siphons in Petri nets based on place constraints," *IEEE Trans. Syst., Man Cybern. A: Syst. Humans*, vol. 35, no. 6, pp. 844–854, Nov. 2005.

[44] J. M. Couvreur, E. Encrenaz, E. Paviot-Adet, D. Poitrenaud, and P. A. Wacrenier, "Data decision diagrams for Petri net analysis," in *Proc. 23rd Int. Conf. Appl. Theory Petri Nets*, 2002, LNCS 2360, pp. 101–120.

[45] J. M. Couvreur and Y. Thierry-Mieg, "Hierarchical decision diagrams to exploit model structure," in *Proc. Formal Tech. Netw. Distrib. Syst.*, 2005, LNCS 3731, pp. 443–457.

[46] D. C. Craig and W. M. Zuberek, "Efficient siphon-based deadlock detection in Petri nets," presented at the 3rd Int. Conf. Comput. Sci. Inf. Syst., Athens, Greece, Jul. 23–26, 2007.

[47] Ph. Darondeau, "Region based synthesis of P/T-nets and its potential applications," in *Proc. 21st Int. Conf. Appl. Theory Petri Nets*, LNCS 1825, 2000, pp. 16–23.

[48] R. David and H. Alla, *Petri Nets and Grafcet*. London, U.K.: Prentice-Hall, 1992.

[49] A. A. Desrochers Ed., *Modeling and Control of Automated Manufacturing Systems*: Piscataway, NJ: IEEE Press, 1989.

[50] A. A. Desrochers and R. Y. AI-Jaar, *Applications of Petri Nets in Manufacturing Systems: Modeling, Control, and Performance Analysis*. New York: IEEE Press, 1994.

[51] F. DiCesare, G. Harhalakis, J. M. Porth, and F. B. Vernadat, *Practice of Petri Nets in Manufacturing*. London, U.K.: Chapman and Hall, 1993.

[52] E. W. Dijkstra, "Cooperating sequential processes," in *Programming Languages*, F. Genuys, Ed., New York: Academic, 1968, pp. 102–110.

[53] M. Dotoli and M. P. Fanti, "Deadlock detection and avoidance strategies for automated storage and retrieval systems," *IEEE Trans. Syst. Man Cybern. C: Appl. Rev.*, vol. 37, no. 4, pp. 541–552, Jul. 2007.

[54] K. A. D'souza and S. K. Khator, "A survey of Petri nets in automated manufacturing systems control," *Comput. Ind. Eng.*, vol. 24, no. 1, pp. 5–16, 1994.

[55] C. Dupont-Gatelmand, "A survey of flexible manufacturing systems," *J. Manuf. Syst.*, vol. 1, no. 1, pp. 1–15, 1982.

[56] (2011). *Engineering Index*, [Online]. Available: http://www.engineeringvillage2.org.cn/

[57] J. Esparza and M. Silva, "Circuits, handles, bridges, and nets," in *Advances in Petri Nets 1990*, LNCS 483, G. Rozenberg, Ed. Berlin, Germany: Springer-Verlag, 1990, pp. 210–242.

[58] J. Ezpeleta, J. M. Couvreur, and M. Silva, "A new technique for finding a generating family of siphons, traps, and st-components: Application to colored Petri nets," in *Advances in Petri Nets*, LNCS 674, G. Rozenberg, Ed. New York: Springer-Verlag, 1993, pp. 126–147.

[59] J. Ezpeleta, J. M. Colom, and J. Martinez, "A Petri net based deadlock prevention policy for flexible manufacturing systems," *IEEE Trans. Robot. Autom.*, vol. 11, no. 2, pp. 173–184, Apr. 1995.

[60] J. Ezpeleta, F. Tricas, F. Garcia-Valles, and J. M. Colom, "A banker's solution for deadlock avoidance in FMS with flexible routing and

multiresource states," *IEEE Trans. Robot. Autom.*, vol. 18, no. 4, pp. 621–625, Aug. 2002.

[61] J. Ezpeleta and L. Recalde, "A deadlock avoidance approach for non-sequential resource allocation systems," *IEEE Trans. Syst., Man, Cybern. A: Syst. Humans*, vol. 34, no. 1, pp. 93–101, Jan. 2004.

[62] A. Fanni, A. Giua, and N. Sanna, "Control and error recovery of Petri net models with event observers," in *Proc. 2nd Int. Work. Manuf. Petri Nets*, Toulouse, France, Jun. 1997, pp. 53–68.

[63] M. P. Fanti, G. Maione, and B. Turchiano, "Digraph-theoretic approach for deadlock detection and recovery in flexible production systems," *Stud. Informat. Control*, vol. 5, no. 4, pp. 373–383, 1996.

[64] M. P. Fanti, G. Maione, and B. Turchiano, "Deadlock detection and recovery in flexible production systems with multiple capacity resources," in *Proc. 8th Mediterranean Electrotechn. Conf.*, Bari, Italy, May 13–16, 1996, pp. 237–241.

[65] M. P. Fanti, B. Maione, S. Mascolo, and B. Turchiano, "Performance of deadlock avoidance algorithms in flexible manufacturing systems," *J. Manuf. Syst*., vol. 15, no. 3, pp. 164–178, 1996.

[66] M. P. Fanti, B. Maione, S. Mascolo, and B. Turchiano, "Event-based feedback control for deadlock avoidance in flexible production systems," *IEEE Trans. Robot. Autom.*, vol. 13, no. 3, pp. 347–363, 1997.

[67] M. P. Fanti, B. Maione, and B. Turchiano, "Event control for deadlock avoidance in production systems with multiple capacity resources," *Stud. Informat. Control*, vol. 7, no. 4, pp. 343–364, 1998.

[68] M. P. Fanti, B. Maione, and B. Turchiano, "Comparing digraph and Petri net approaches to deadlock avoidance in FMS," *IEEE Trans. Syst., Man Cybern. B: Cybern.*, vol. 30, no. 5, pp. 783–798, Oct. 2000.

[69] M. P. Fanti, G. Maione, and B. Turchiano, "Distributed event-control for deadlock avoidance in automated manufacturing systems," *Int. J. Prod. Res.*, vol. 39, no. 9, pp. 1993–2021, 2001.

[70] M. P. Fanti, "Event-based controller to avoid deadlock and collisions in zone control AGVs," *Int. J. Prod. Res.*, vol. 40, no. 6, pp. 1453–1478, 2002.

[71] M. P. Fanti, G. Maione, and B. Turchiano, "Design of supervisors to avoid deadlock in flexible assembly systems," *Int. J. Flexible Manuf. Syst.*, vol. 14. no. 2, pp. 157–175, 2002.

[72] M. P. Fanti and M. C. Zhou, "Deadlock control methods in automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A: Syst. Humans*, vol. 34, no. 1, pp. 5–22, Jan. 2004.

[73] M. P. Fanti, A. Giua, and C. Seatzu, "Monitor design for colored Petri nets: An application to deadlock prevention in railway networks," *Control Eng. Practice*, vol. 14, no. 10, pp. 1231–1247, 2006.

[74] F. García-Vallés and J. M. Colom, "Implicit places in net systems," in *Proc. 8th Int. Workshop Petri Nets Perform. Models*, Zaragoza, Spain, 1999, pp. 104–113.

[75] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: Freeman, 1979.

[76] A. Ghaffari, N. Nidhal, and X. L. Xie, "Design of a live and maximally permissive Petri net controller using the theory of regions," *IEEE Trans. Robot. Autom.*, vol. 19, no. 2, pp. 137–142, Feb. 2003.

[77] A. Giua, F. DiCesare, and M. Silva, "Generalized mutual exclusion constraints on nets with uncontrollable transitions," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Chicago, IL, 1992, pp. 974–979.

[78] A. Giua and F. DiCesare, "On the existence of Petri net supervisors," in *Proc. IEEE 31st Conf. Decis. Control*, Tucson, AZ, 1992, pp. 3380–3385.

[79] A. Giua, F. DiCesare, and M. Silva, "Petri net supervisor for generalized mutual exclusion constraints," in *Proc. 12th IFAC World Congr.*, Sidney, Australia, Jul. 1993, vol. 1, pp. 267–270.

[80] A. Giua and C. Seatzu, "Obervability of place/transition nets," *IEEE Trans. Automat. Control*, vol. 47, no. 9, pp. 1424–1437, Sep. 2002.

[81] A. Giua and C. Seatzu, "Liveness enforcing supervisors for railway networks using ES^2PR Petri nets," in *Proc. 6th Int. Workshop Discrete Event Syst.*, Zaragoza, Spain, 2002, pp. 55–60.

[82] A. Giua and X. L. Xie, "Control of safe ordinary Petri nets using unfolding," *Discrete Event Dyn. Syst.*, vol. 15, no. 3, pp. 349–373, 2005.

[83] A. Giua and C. Seatzu, "A systems theory view of Petri nets," in *Advances in Control Theory and Applications*, LNCIS 353, C. Bonivento, A. Isidori, L. Marconi, and C. Rossi, Eds. Berlin, Germany: Springer-Verlag, 2007, pp. 99–127.

[84] A. Giua and C. Seatzu, "Modeling and supervisory control of railway networks using Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 5, no. 3, pp. 431–445, Jul. 2008.

[85] E. M. Gold, "Deadlock predication: Easy and difficult cases," *SIAM J. Comput.*, vol. 7, no. 3, pp. 320–336, 1978.

[86] J. W. Guo and Z. W. Li, "A deadlock prevention approach for a class of timed Petri nets using elementary siphons," *Asian J. Control*, vol. 12, no. 3, pp. 347–363, 2010.

[87] A. Haberman, "Prevention of system deadlocks," *Commun. ACM*, vol. 12, no. 7, pp. 373–377, 1969.

[88] A. Hamez, Y. Thierry-Mieg, F. Kordon, "Building efficient model checkers using hierarchical set decision diagrams and automatic saturation," *Fundam. Informat.*, vol. 94, nos. 3–4, pp. 413–437, 2009.

[89] K. X. He and L. D. Lemmon, "On the transformation of maximally permissive marking-based liveness enforcing supervisors into monitor supervisors," in *Proc. IEEE 39th Conf. Decis. Control*, Sydney, NSW, Australia, Dec. 12–15, 2002, vol. 3, 2657–2662.

[90] K. X. He and M. D. Lemmon, "Liveness verification of discrete-event systems modeled by n-safe ordinary Petri nets," in *Proc. 21st Int. Conf. Appl. Theory Petri Nets*, LNCS 1825, 2000, pp. 227–243.

[91] K. X. He and M. D. Lemmon, "Liveness-enforcing supervision of bounded ordinary Petri nets using partial order methods," *IEEE Trans. Automat. Control*, vol. 47, no. 7, pp. 1042–1055, Jul. 2002.

[92] L. E. Holloway, B. H. Krogh, and A. Giua, "A survey of Petri net methods for controlled discrete event systems," *Discrete Event Dyn. Syst.: Theory Appl.*, vol. 7, no. 2, pp. 151–190, 1997.

[93] R. Holt, "Some deadlock properties of computer systems," *ACM Comput. Surv.*, vol. 4, no. 3, pp. 179–196, 1972.

[94] B. Hruz and M. C. Zhou, *Modeling and Control of Discrete Event Dynamic Systems: With Petri Nets and Other Tools*. London, U.K.: Springer-Verlag, 2007.

[95] F. S. Hsieh and S. C. Chang, "Dispatching-driven deadlock avoidance controller synthesis for flexible manufacturing systems," *IEEE Trans. Robot. Autom.*, vol. 10, no. 2, pp. 196–209, Apr. 1994.

[96] F. S. Hsieh, "Fault-tolerant deadlock avoidance algorithm for assembly processes," *IEEE Trans. Syst., Man, Cybern. A: Syst. Humans*, vol. 34, no. 1, pp. 65–79, Jan. 2004.

[97] (2010). [Online]. Available: http://en.wikipedia.org/wiki/Ostrich_algorithm

[98] H. S. Hu and Z. W. Li, "An optimal-elementary-siphons based iterative deadlock prevention policy for flexible manufacturing systems," *Int. J. Adv. Manuf. Technol.*, vol. 38, nos. 3–4, pp. 309–320, Aug. 2008.

[99] H. S. Hu, M. C. Zhou, and Z. W. Li, "Liveness enforcing supervision of video streaming systems using nonsequential Petri nets," *IEEE Trans. Multimedia*, vol. 11, no. 8, pp. 1457–1465, Dec. 2009.

[100] H. S. Hu and Z. W. Li, "Local and global deadlock prevention policies for resource allocation systems using partially generated reachability graphs," *Comput. Ind. Eng.*, vol. 57, no. 4, pp. 1168–1181, Nov. 2009.

[101] H. S. Hu and Z. W. Li, "Efficient deadlock prevention policy in automated manufacturing systems using exhausted resources," *Int. J. Adv. Manuf. Technol.*, vol. 40, nos. 5–6, pp. 566–571, Jan. 2009.

[102] H. S. Hu and Z. W. Li, "Synthesis of liveness enforcing supervisor for automated manufacturing systems using insufficiently marked siphons," *J. Intell. Manuf.*, vol. 21, no. 4, pp. 555–567, Aug. 2010.

[103] H. S. Hu, M. C. Zhou, and Z. W. Li, "Algebraic synthesis of timed supervisor for automated manufacturing systems using Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 3, pp. 549–557, Jul. 2010.

[104] H. S. Hu, M. C. Zhou, and Z. W. Li, "Low-cost and high-performance supervision in ratio-enforced automated manufacturing systems using timed Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 4, pp. 933–944, Oct. 2010.

[105] H. S. Hu, M. C. Zhou, and Z. W. Li, "Supervisor design to enforce production ratio and absence of deadlock in automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A: Syst. Humans*, vol. 41, no. 2, pp. 201–212, Mar. 2011.

[106] Y. S. Huang, M. D. Jeng, X. L. Xie, and S. L. Chung, "Deadlock prevention policy based on Petri nets and siphons," *Int. J. Prod. Res.*, vol. 39, no. 1, pp. 283–305, 2001.

[107] Y. S. Huang, M. D. Jeng, X. L. Xie, and S. L. Chung, "A deadlock prevention policy for flexible manufacturing systems using siphons," in *Proc. IEEE Int. Conf. Robot. Autom.*, Seoul, Korea, 2001, pp. 541–546.

[108] Y. S. Huang, M. D. Jeng, X. L. Xie, and D. H. Chung, "Siphon-based deadlock prevention policy for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A: Syst. Humans*, vol. 36, no. 6, pp. 2152–2160, 2006.

[109] M. V. Iordache and P. J. Antsaklis, "Generalized conditions for liveness enforcement and deadlock prevention in Petri nets," in *Proc. 22nd Int. Conf. Applicat. Theory Petri Nets*, LNCS 2075, 2001, pp. 184–203.

[110] M. V. Iordache, J. O. Moody, and P. J. Antsaklis, "Synthesis of deadlock prevention supervisors using Petri nets," *IEEE Trans. Robot. Autom.*, vol. 18, no. 1, pp. 59–68, Feb. 2002.
[111] M. V. Iordache, "Methods for the supervisory control of concurrent systems based on petri net abstractions," Ph.D. dissertation, Univ. of Notre Dame, Notre Dame, IN, 2003.
[112] M. V. Iordache and P. J. Antsaklis, "Synthesis of supervisors enforcing general linear vector constraints in Petri nets," *IEEE Trans. Automat. Control*, vol. 48, no. 11, pp. 2036–2039, Nov. 2003.
[113] M. V. Iordache and P. J. Antsaklis, *Supervisory Control of Concurrent Systems: A Petri Net Structural Approach*. Berlin, Germany: Springer-Verlag, 2006.
[114] M. V. Iordache and P. J. Antsaklis, "Supervision based on place invariants: A survey," *Discrete Event Dyn. Syst.: Theory Appl.*, vol. 16, no. 4, pp. 451–492, 2006.
[115] S. S. Isloor and T. A. Marsland, "The deadlock problem: An overview," *Computer*, vol. 13, no. 9, pp. 58–77, 1980.
[116] M. D. Jeng and F. DiCesare, "A review of synthesis techniques for Petri nets with applications to automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 1, pp. 301–312, Jan./Feb. 1993.
[117] M. D. Jeng and F. DiCesare, "Synthesis using resource control nets for modeling shared-resource systems," *IEEE Trans. Robot. Autom.*, vol. 11, no. 3, pp. 317–327, Jun. 1995.
[118] M. D. Jeng, "A Petri net synthesis theory for modeling flexible manufacturing systems," *IEEE Trans. Syst., Man Cybern. B: Cybern.*, vol. 27, no. 2, pp. 169–183, Apr. 1997.
[119] M. D. Jeng and X. L. Xie, "Analysis of modularly composed nets by siphons," *IEEE Trans. Syst., Man, Cybern. A: Syst. Humans*, vol. 29, no. 4, pp. 399–406, Jul. 1999.
[120] M. D. Jeng and X. L. Xie, "Modeling and analysis of semiconductor manufacturing systems with degraded behavior using Petri nets and siphons," *IEEE Trans. Robot. Autom.*, vol. 17, no. 5, pp. 576–588, Oct. 2001.
[121] M. D. Jeng, "Modeling, specification and analysis of manufacturing systems," *Int. J. Prod. Res.*, vol. 39, no. 2, pp. 225–253, 2001 (special issue).
[122] M. D. Jeng, X. L. Xie, and M. Y. Peng, "Process nets with resources for manufacturing modeling and their analysis," *IEEE Trans. Robot. Autom.*, vol. 18, no. 6, pp. 875–889, Dec. 2002.
[123] M. D. Jeng, X. L. Xie, and S. L. Chung, "ERCN* merged nets for modeling degraded behavior and parallel processes in semiconductor manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A: Syst. Humans*, vol. 34, no. 1, pp. 102–112, Jan. 2004.
[124] M. D. Jeng and X. L. Xie, "Deadlock detection and prevention of automated manufacturing systems using Petri nets and siphons," in *Deadlock Resolution in Computer-Integrated Systems*, M. C. Zhou and M. P. Fanti, Eds. New York: Marcel-Dekker, 2005, pp. 233–281.
[125] M. D. Jeng and X. L. Xie, "Discrete event system technique for CIM," *Int. J. Comput. Integr. Manuf.*, vol. 18, nos. 2–3, 2005 (special issue).
[126] M. D. Jeng and M. C. Zhou, "Engineering applications of Petri Nets," *IEEE Trans. Syst., Man, Cybern. C-Appl. Rev.*, vol. 37, no. 4, 2007 (special issue).
[127] K. Jensen, *Colored Petri Nets: Basic Concepts, Analysis Methods, and Practical Use*. New York: Springer-Verlag, 1994.
[128] M. Khalgui and H. M. Hanisch, "Reconfiguration protocol for multi-agent control software architectures," *IEEE Trans. Syst., Man Cybern. C: Appl. Rev.*, vol. 41, no. 1, pp. 70–80, 2011.
[129] C. O. Kim and S. S. Kim, "An efficient real-time deadlock-free control algorithm for automated manufacturing systems," *Int. J. Prod. Res.*, vol. 35, no. 6, pp. 1545–1560, 1997.
[130] S. Kumagai (Ed.), "Application of Petri Nets to Concurrent System Design," *IEICE Trans. Fund. Electron., Commun., Comput. Sci.*, vol. 75, no. 10, 1992 (special issue).
[131] T. Kumaran, W. Chang, H. Cho, and A. Wysk, "A structured approach to deadlock detection, avoidance and resolution in flexible manufacturing systems," *Int. J. Prod. Res.*, vol. 32, no. 10, pp. 2361–2379, 1994.
[132] K. Lautenbach, "Linear algebraic calculation of deadlocks and traps," in *Concurrency and Nets*, K. Voss, H. J. Genrich and G. Rozenberg, Eds. New York: Springer-Verlag, 1987, pp. 315–336.
[133] K. Lautenbach and H. Ridder, "Liveness in bounded Petri nets which are covered by T-invariants," in *Proc. 13th Int. Conf. Applicat. Theory Petri Nets*, LNCS 815, 1994, pp. 358–375.
[134] K. Lautenbach and H. Ridder, "The linear algebra of deadlock avoidance—A Petri net approach," Inst. of Soft. Technol., Univ. of Koblenz-Landau, Koblenz, Germany, Tech. Rep. 25-1996, 1996.
[135] M. A. Lawley, S. A. Reveliotis, and P. M. Ferreira, "Design guidelines for deadlock-handling strategies in flexible manufacturing systems," *Int. J. Flexible Manuf. Syst.*, vol. 9, no. 1, pp. 5–30, 1997.
[136] 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 Trans.*, vol. 29, no. 10, pp. 877–887, 1997.
[137] 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 Trans.*, vol. 29, no. 10, pp. 889–899, 1997.
[138] M. A. Lawley, S. A. Reveliotis, and P. M. Ferreira, "A correct and scalable deadlock avoidance policy for flexible manufacturing systems," *IEEE Trans. Robot. Autom.*, vol. 14, no. 5, pp. 796–809, Oct. 1998.
[139] M. A. Lawley, S. A. Reveliotis, and P. M. Ferreira, "The application and evaluation of banker's algorithm for deadlock-free buffer space allocation in flexible manufacturing systems," *Int. J. Flexible Manuf. Syst.*, vol. 10, no. 1, pp. 73–100, 1998.
[140] M. A. Lawley, "Deadlock avoidance for production systems with flexible routing," *IEEE Trans. Robot. Autom.*, vol. 15, no. 3, pp. 497–509, Jun. 1999.
[141] M. A. Lawley, "Integrating flexible routing and algebraic deadlock avoidance policies in automated manufacturing systems," *Int. J. Prod. Res.*, vol. 38, no. 13, pp. 2931–2950, 2000.
[142] M. A. Lawley and S. A. Reveliotis, "Deadlock avoidance for sequential resource allocation systems: Hard and easy cases," *Int. J. Flexible Manuf. Syst.*, vol. 13, no. 4, pp. 385–404, 2001.
[143] M. A. Lawley and W. Sulistyono, "Robust supervisory control policies for manufacturing systems with unreliable resources," *IEEE Trans. Robot. Autom.*, vol. 18, no. 3, pp. 346–359, Jun. 2002.
[144] Y. T. Leung and G. J. Sheen, "Resolving deadlocks in flexible manufacturing cells," *J. Manuf. Syst.*, vol. 12, no. 4, pp. 291–304, 1993.
[145] S. Y. Li and Z. W. Li, "Solving siphons with the minimal cardinality in Petri nets and its application to deadlock control," *Int. J. Prod. Res.*, 2011, to be published.
[146] S. Y. Li, Z. W. Li, and H. S. Hu, "Siphon extraction for deadlock control in flexible manufacturing systems by using Petri nets," *Int. J. Comput. Integr. Manuf.*, vol. 24, no. 8, pp. 710–725, 2011.
[147] S. Y. Li and Z. W. Li, "Structure reduction of liveness-enforcing Petri nets using mixed integer programming," *Asian J. Control*, [Online]. DOI: 10.1002/asjc.342, 2011.
[148] Z. W. Li and M. C. Zhou, "Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A: Syst. Humans*, vol. 34, no. 1, pp. 38–51, Jan. 2004.
[149] Z. W. Li and M. C. Zhou, "Comparison of two deadlock prevention methods for different-size flexible manufacturing systems," *Int. J. Intell. Control Syst.*, vol. 10, no. 3, pp. 235–243, 2005.
[150] Z. W. Li and M. C. Zhou, "Two-stage method for synthesizing liveness-enforcing supervisors for flexible manufacturing systems using Petri nets," *IEEE Trans. Ind. Informat.*, vol. 2, no. 4, pp. 313–325, Nov. 2006.
[151] Z. W. Li and M. C. Zhou, "Clarifications on the definitions of elementary siphons of Petri nets," *IEEE Trans. Syst., Man, Cybern. A: Syst. Humans*. vol. 36, no. 6, pp. 1277–1279, Nov. 2006.
[152] Z. W. Li and N. Wei, "Deadlock control of flexible manufacturing systems via invariant-controlled elementary siphons of Petri nets," *Int. J. Adv. Manuf. Technol.*, vol. 33, nos. 1–2, pp. 24–35, 2007.
[153] Z. W. Li, H. S. Hu, and A. R. Wang, "Design of liveness-enforcing supervisors for flexible manufacturing systems using Petri nets," *IEEE Trans. Syst., Man, Cybern. C: Appl. Rev.*, vol. 37, no. 4, pp. 517–526, Jul. 2007.
[154] Z. W. Li and D. Liu, "A correct minimal siphons extraction algorithm from a maximal unmarked siphon of a Petri net," *Int. J. Prod. Res.*, vol. 45, no. 9, pp. 2161–2165, 2007.
[155] Z. W. Li, M. C. Zhou, and M. Uzam, "Deadlock control policy for a class of Petri nets without complete siphon enumeration," *IET Control Theory Appl.*, vo1. 1, no. 6, pp. 1594–1605, 2007.
[156] Z. W. Li, M. C. Zhou, and N. Q. Wu, "A survey and comparison of Petri net-based deadlock prevention policies for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. C: Appl. Rev.*, vol. 38, no. 2, pp. 172–188, Mar. 2008.
[157] Z. W. Li and M. C. Zhou, "On siphon computation for deadlock control in a class of Petri nets," *IEEE Trans. Syst., Man, Cybern. A: Syst. Human Beings*, vol. 38, no. 3, pp. 667–679, May 2008.

[158] Z. W. Li and M. Zhao, "On controllability of dependent siphons for deadlock prevention in generalized Petri nets," *IEEE Trans. Syst., Man, Cybern. A: Syst. Humans*, vol. 38, no. 2, pp. 369–384, Mar. 2008.

[159] 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 Trans. Autom. Sci. Eng.*, vol. 5, no. 1, pp. 182–188, Jan. 2008.

[160] Z. W. Li and M. C. Zhou, "Control of elementary and dependent siphons of Petri nets and their application," *IEEE Trans. Syst., Man, Cybern. A: Syst. Human*, vol. 38, no. 1, pp. 133–148, Jan. 2008.

[161] Z. W. Li and M. C. Zhou, *Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach*. New York: Springer-Verlag, 2009.

[162] Z. W. Li and M. C. Zhou, "A polynomial-complexity approach to decide the existence of a maximally permissive Petri net supervisor using elementary siphons," in *Proc. IEEE Int. Conf. Netw., Sens. Control*, 2009, pp. 608–613.

[163] Z. W. Li, S. Zhu, and M. C. Zhou, "A divide-and-conquer strategy to deadlock prevention in flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. C: Appl. Rev.*, vol. 39, no. 2, pp. 156–169, Mar. 2009.

[164] Z. W. Li and H. S. Hu, "On systematic methods to remove redundant monitors from liveness-enforcing net supervisors," *Comput. Ind. Eng.*, vol. 56, no. 1, pp. 53–62, 2009.

[165] Z. W. Li, M. Qin, and S. Zhu, "Identification of controllable transitions to decide the existence of an optimal liveness-enforcing supervisor for a class of Petri nets," *Trans. Inst. Meas. Control*, vol. 33, nos. 3-4, pp. 406–421, 2011.

[166] Z. W. Li and M. Shpitalni, "Smart deadlock prevention policy for flexible manufacturing systems using Petri nets," *IET Control Theory Appl.*, vol. 3, no. 3, pp. 362–374, 2009.

[167] Z. W. Li and M. C. Zhou, "Synthesis of structurally simple supervisors enforcing generalized mutual exclusion constraints in Petri nets," *IEEE Trans. Syst., Man Cybern. C: Appl. Rev.*, vol. 40, no. 3, pp. 330–340, May 2010.

[168] D. Liu, X. Li, and Z. W. Li. (2009). FANM: Extraction of a minimal siphon from a maxmial one [Online]. Available: http//www.google.com/zhiwulixidian.html.

[169] D. Liu, Z. W. Li, and M. C. Zhou, "Liveness of an extended S^3PR," *Automatica*, vol. 46, pp. 1008–1018, 2010.

[170] G. J. Liu, C. J. Jiang, Z. H. Wu, and L. J. Chen, "A live subclass of Petri nets and their application in modeling flexible manufacturing systems," *Int. J. Adv. Manuf. Technol.*, vol. 41, nos. 1–2, pp. 66–74, 2009.

[171] G. J. Liu, C. J. Jiang, L. J. Chen, and Z. H. Wu, "Two types of extended RSNBs and their application in modeling flexible manufacturing systems," *Int. J. Adv. Manuf. Technol.*, vol. 45, nos. 5–6, pp. 573–582, 2009.

[172] G. J. Liu, C. J. Jiang, and M. C. Zhou, "Two simple deadlock prevention policies for S3PR based on key-resource/operation-place pairs," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 4, pp. 945–957, Jun. 2010.

[173] G. Y. Liu, Z. W. Li, and C. F. Zhong, "New controllability condition for siphons in a class of generalized Petri nets," *IET Control Theory Appl.*, vol. 4, no. 5, pp. 854–864, 2010.

[174] 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 Appl.*, vol. 4, no. 12, pp. 2867–2878, 2010.

[175] J. P. López-Grao and J. M. Colom, "Lender processes cpmpeting for shared resources: Beyond the S^4PR paradigm," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Taipei, China, Oct. 8–11 2011, pp. 3052–3059.

[176] J. P. López-Grao and J. M. Colom, "The resource allocation problem in software applications: A petri net perspective," in *Proc. Int. Workshop Petri Nets Softw. Eng.*, Braga, Portugal, 2010, pp. 7–21.

[177] G. Maione and F. DiCesare, "Hybrid Petri net and digraph approach for deadlock prevention in automated manufacturing systems," *Int. J. Prod. Res.*, vol. 43, no. 24, pp. 5131–5159, 2005.

[178] K. McMillan, "Using unfoldings to avoid the state explosion problem in the verification of asynchronous circuits," in *Lecture Notes in Computer Science*, vol. 663, B. V. Bochmann and D. K. Probst, Eds. Berlin, Germany: Springer, 1992, pp. 164–177.

[179] K. McMillan, *Symbolic Model Checking*, Norwell, MA: Kluwer, 1993.

[180] A. S. Miner and G. Ciardo, "Efficient reachability set generation and storage using decision diagrams," in *Proc. 20th Int. Conf. Appl. Theory Petri Nets*, vol. LNCS-1639, S. Donatelli and H. C. M. Kleijn, Eds. Berlin, Germany: Springer-Verlag, 1999, pp. 6–25.

[181] T. Minoura, "Deadlock avoidance revisited," *J. ACM*, vol. 29, no. 4, pp. 1023–1048, 1982.

[182] J. O. Moody and P. J. Antsaklis, "Supervisory control of Petri nets with uncontrollable/unobservable transitions," in *Proc. 35th Conf. Decis. Control*, vol. 4, Kobe, Japan, 1996, pp. 4433–4438.

[183] J. O. Moody and P. J. Antsaklis, "Petri net supervisors for DES with uncontrollable and unobservable transitions," *IEEE Trans. Automat. Control*, vol. 45, no. 3, pp. 462–476, Mar. 2000.

[184] K. E. Moore and S. M. Guptats, "Petri net models of flexible and automated manufacturing systems: A survey," *Int. J. Prod. Res.*, vol. 34, no. 11, pp. 3001–3035, 1996.

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

[186] G. Newton, "Deadlock prevention, detection, and resolution: An annotated bibliography," *ACM SIGOPS Operat. Syst. Rev.*, vol. 13, no. 2, pp. 33–44, 1979.

[187] N. G. Odrey and G. Mejía, "A re-configurable multi-agent system architecture for error recovery in production systems," *Robot. Comput.-Integr. Manuf.*, vol. 19, nos. 1–2, pp. 35–43, 2003.

[188] N. G. Odrey and G. Mejía, "An augmented Petri net approach for error recovery in manufacturing systems control," *Robot. Comput.-Integr. Manuf.*, vol. 21, nos. 4–5, pp. 346–354, 2005.

[189] J. Park, "Structural analysis and control of resource allocation systems using petri nets," Ph.D. dissertation, Georgia Inst. Technol., Atlanta, GA, 2000.

[190] J. Park and S. A. Reveliotis, "Algebraic synthesis of efficient deadlock avoidance policies for sequential resource allocation systems," *IEEE Trans. Robot. Autom.*, vol. 16, no. 2, pp. 190–195, Apr. 2000.

[191] J. Park and S. A. Reveliotis, "Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings," *IEEE Trans. Automat. Control*, vol. 46, no. 10, pp. 1572–1583, Oct. 2001.

[192] E. Pastor, J. Cortadella, and O. Roig, "Symbolic analysis of bounded Petri nets," *IEEE Trans. Comput.*, vol. 50, no. 5, pp. 432–448, May 2001.

[193] J. L. Peterson, "Petri nets," *Comput. Surv.*, vol. 9, no. 3, pp. 223–252, 1977.

[194] J. L. Peterson, *Petri Net Theory and the Modeling of Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1981.

[195] L. Piroddi, R. Cordone, and I. Fumagalli, "Selective siphon control for deadlock prevention in Petri nets," *IEEE Trans. Syst., Man, Cybern. A: Syst. Humans*, vol. 38, no. 6, pp. 1337–1348, Nov. 2008.

[196] L. Piroddi, R. Cordone, and I. Fumagalli, "Combined siphon and marking generation for deadlock prevention in Petri nets," *IEEE Trans. Syst., Man, Cybern. A: Syst. Humans*, vol. 39, no. 3, pp. 650–661, May 2009.

[197] J. M. Proth and X. L. Xie, *Petri Nets, A Tool for Design and Management of Manufacturing Systems*. New York: Wiley, 1996.

[198] M. Qin, Z. W. Li, M. Khalgui, and O. Mosbahi, "On applicability of deadlock prevention policies with uncontrollable and unobservable transitions," *Int. J. Innovat. Comput., Inf., Control*, vol. 7, no. 6, pp. 4115–4127, 2011.

[199] P. J. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proc. IEEE*, vol. 77, no. 1, pp. 81–89, Jan. 1989.

[200] S. Ramaswamy, K. P. Valavanis, and S. Barber, "Petri net extensions for the development of MIMO net models of automated manufacturing systems," *J. Manuf. Syst.*, vol. 16, no. 3, pp. 175–191, 1997.

[201] L. Recalde, E. Teruel, and M. Silva, "Improving the decision power of rank theorems," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Orlando, FL, 1997, pp. 3768–3773.

[202] W. Reisig, *Petri Nets: An Introduction*. New York: Springer-Verlag, 1985.

[203] S. A. Reveliotis and P. M. Ferreira, "Deadlock avoidance policies for automated manufacturing cells," *IEEE Trans. Robot. Autom.*, vol. 12, no. 6, pp. 845–857, Dec. 1996.

[204] S. A. Reveliotis, M. A. Lawley, and P. M. Ferreira, "Polynomial-complexity deadlock avoidance policies for sequential resource allocation systems," *IEEE Trans. Automat. Control*, vol. 42, no. 10, pp. 1344–1357, Oct. 1997.

[205] S. A. Reveliotis, "Liveness enforcing supervision for sequential resource allocation systems: State of the art and open issues," in *Synthesis and Control of Discrete Event Systems*, B. Caillaud, Ph. Darondeau, L. Lavagno, and X. L. Xie, Eds. Dordrecht, The Netherlands: Kluwer, 2002, pp. 203–212.

[206] S. A. Reveliotis, "On the siphon-based characterization of liveness in sequential resource allocation systems," in *Proceedings of the 24th*

International Conference on Applications and Theory of Petri Nets, LNCS, vol. 2679, W. M. P. van der Aalst and E. Best, Eds. Berlin, Germany: Springer-Verlag, 2003, pp. 241–255.

[207] S. A. Reveliotis, *Real-time Management of Resource Allocation Systems: A Discrete Event Systems Approach*. New York: Springer-Verlag, 2005.

[208] S. A. Reveliotis, "Algebraic deadlock avoidance policies for sequential resource allocation systems," in *Facility Logistics: Approaches and Solutions to Next Generation Challenges*, M. Lahmar, Ed. New York: Taylor & Francis, 2007, pp., 235–289.

[209] E. Roszkowska and R. Wojcik, "Problems of process flow feasibility in FAS," in *CIM in Process and Manufacturing Industries*. Oxford, U.K.: Pergamon Press, pp. 115C120, 1993.

[210] E. Roszkowska and J. Jentink, "Minimal restrictive deadlock avoidance in FMSs," in *Proc. Eur. Control Conf.*, 1993, vol. 2, pp. 530–534.

[211] M. Singhal, "Deadlock detection in distributed systems," *IEEE Comput.*, vol. 22, no. 11, pp. 37–48, Nov. 1989.

[212] M. Silva and R. Valette, "Petri Nets and Flexible Manufacturing," in *Advances in Petri Nets*, Lecture Notes in Computer Science, vol. 424, G. Rozenberg, Ed. Heidelberg, Germany: Springer-Verlag, 1989, pp. 374–417.

[213] M. Silva, R. Valette, and K. Takahashi, (Eds)., *Proceedings of the 1st Workshop on Manufacturing and Petri Nets 17th Int. Conf. Appl. Theory of Petri Nets*, Osaka, Japan, Jun. 24–28, 1996.

[214] M. Silva and E. Teruel, "A systems theory perspective of discrete event dynamic systems: The Petri net paradigm," in *Proc. Symp. Discrete Events Manuf. Syst.*, Lille, France, 1996, pp. 1–12.

[215] M. Silva, E. Teruel, R. Valette, and H. Pingaud, "Petri Nets and Production Systems," in *Lectures on Petri Nets II: Applications*, *Lecture Notes in Computer Science,* vol. 1492, W. Reisig G. Rozenberg, Eds. Berlin, Germany: Springer 1998, pp. 85–124.

[216] R. S. Sreenivas,"On Commoner's liveness theorem and supervisory policies that enforce liveness in free-choice Petri nets," *Syst. Control Lett.*, vol. 31, no. 1, pp. 41–48, 1997.

[217] R. S. Sreenivas,"On the existence of supervisory control policies that enforce liveness in discrete-event dynamic systems modeled by controlled Petri nets," *IEEE Trans. Automat. Control*, vol. 42, no. 7, pp. 928–945, Jul. 1997.

[218] R. S. Sreenivas,"On supervisory policies that enforce liveness in completely controlled Petri nets with directed cut-places and cut-transitions," *IEEE Trans. Automat. Control*, vol. 44, no. 6, pp. 1221–1225, Jun. 1999.

[219] A. Taubin, A. Kondratyev, and M. Kishinevsky, "Deadlock prevention using Petri nets and their unfoldings," *Int. J. Adv. Manuf. Technol.*, vol. 14, no. 10, pp. 750–759, 1998.

[220] A. J. C. Trappey, D. W. Hsiao, and L. Ma, "Maintenance chain integration using Petri-net enabled multi-agent system modeling and implementation approach," *IEEE Trans. Syst., Man Cybern. C: Appl. Rev.*, vol. 41, no. 3, pp. 306–315, 2011.

[221] F. Tricas and J. Martinez, "An extension of the liveness theory for concurrent sequential processes competing for shared resources," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Vancouver, BC, Canada, 1995, pp. 3035–3040.

[222] F. Tricas, F. Garcia-Vallès, J. M. Colom, and J. Ezpeleta, "An iterative method for deadlock prevention in FMSs," in *Proc. 5th Workshop Discrete Event Syst.*, R. Boel and G. Stremersch, Eds., Ghent, Belgium, 2000, pp. 139–148.

[223] F. Tricas, *Deadlock Analysis, Prevention and Avoidance in Sequential Resource Allocation Systems*, Ph.D. dissertation, Univ. Zaragoza, Zaragoza, Spain, 2003.

[224] F. Tricas and J. Ezpeleta, "Some results on siphon computation for deadlock prevention in resource allocation systems modeled with Petri nets," in *Proc. IEEE Conf. Emerg. Technol. Factory Autom.*, Lisbon, Portugal, 2003, pp. 322–329.

[225] F. Tricas, F. Garcia-Vallés, J. M. Colom, and J. Ezpeleta, "Using linear programming and the Petri net structure for deadlock prevention in sequential resource allocation systems," in *XIII Jornadas de Concurrencia y Sistemas Distribuidos*, Madrid, Spain: Thomson Paraninfo, 2005, pp. 65–77.

[226] F. Tricas and J. Ezpeleta, "Computing minimal siphons in Petri net models of resource allocation systems: A parallel solution," *IEEE Trans. Syst., Man, Cybern. A: Syst. Humans*, vol. 36, no. 3, pp. 532–539, May 2006.

[227] M. Uzam, "An optimal deadlock prevention policy for flexible manufacturing systems using Petri net models with resources and the theory of regions," *Int. J. Adv. Manuf. Technol.*, vol. 19, no. 3, pp. 192–208, 2002.

[228] M. Uzam, "The use of the Petri net reduction approach for an optimal deadlock prevention policy for flexible manufacturing systems," *Int. J. Adv. Manuf. Technol.*, vol. 23, nos. 3-4, pp. 204–219, 2004.

[229] M. Uzam and M. C. Zhou, "An improved iterative synthesis method for liveness enforcing supervisors of flexible manufacturing systems," *Int. J. Prod. Res.*, vol. 44, no. 10, pp. 1987–2030, 2006.

[230] M. Uzam and M. C. Zhou, "An iterative synthesis approach to Petri net based deadlock prevention policy for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A: Syst. Humans*, vol. 37, no. 3, pp. 362–371, May 2007.

[231] 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," *Int. J. Adv. Manuf. Technol.*, vol. 35, no. 1–2, pp. 150–168, 2007.

[232] A. Valmari, "Stubborn sets for reduced state space generation," in *Advances in Petri Nets 1990*, *Lecture Notes in Computer Science,* G. Rozenberg (Ed.), vol. 483, Berlin, Germany: Springer-Verlag, pp. 491–515, 1991.

[233] A. Valmari, "A stubborn attach on state explosion," *Formal Methods Syst. Design*, vol. 1, no. 4, pp. 297–322, 1992.

[234] K. Varpaaniemi, "Efficient detection of deadlocks in Petri nets," Digital Syst. Lab., Helsinki Univ. Technol., Espoo, Finland, Res. Rep. A26, Oct. 1993.

[235] N. Viswanadham, Y. Narahari, and T. Johnson, "Deadlock prevention and deadlock avoidance in flexible manufacturing systems using Petri net models," *IEEE Trans. Robot. Autom.*, vol. 6, no. 6, pp. 713–723, Dec. 1990.

[236] N. Viswanadham and Y. Narahari, *Performance Modelling of Automated Manufacturing Systems*. Englewood Cliffs, NJ: Pretence-Hall, 1992.

[237] A. R. Wang, Z. W. Li, J. Y. Jian, and M. C. Zhou, "An effective algorithm to find elementary siphons in a class of Petri nets," *IEEE Trans. Syst., Man, Cybern. A: Syst. Humans*, vol. 39, no. 4, pp. 912–923, Jul. 2009.

[238] A. R. Wang, Z. W. Li, and J. Y. Jia, "Efficient computation of strict minimal siphons for a class of Petri nets models of automated manufacturing systems," *Trans. Inst. Meas. Control*, vol. 33, no. 1, pp. 182–201, 2011.

[239] N. Wei and Z. W. Li, "On the suboptimal liveness-enforcing supervisors based on Petri net structural analysis and the theory of regions," *Int. J. Adv. Manuf. Technol.*, vol. 38, nos. 1-2, pp. 195–204, 2008.

[240] W. L. Winston and M. Venkataramanan, *Introduction to Mathematical Programming*, Belmont, CA: Duxbury Resource Center, 2002.

[241] W. M. Wonham. (2009). Supervisory control of discrete event systems' [Online]. Available: http://www.control.toronto.edu/people/profs/wonham/wonham.html

[242] N. Q. Wu, "Necessary and sufficient conditions for deadlock-free operation in flexible manufacturing systems using a colored Petri net model," *IEEE Trans. Syst., Man, Cybern. C: Appl. Rev.*, vol. 29, no. 2, pp. 192–204, May 1999.

[243] N. Q. Wu and M. C. Zhou, "Avoiding deadlock and reducing starvation and blocking in automated manufacturing systems," *IEEE Trans. Robot. Autom.*, vol. 17, no. 5, pp. 657–668, Oct. 2001.

[244] N. Q. Wu, "Deadlock avoidance in AGV system using colored Petri net model," *Int. J. Prod. Res.*, vol. 40, no. 1, pp. 223–238, 2002.

[245] N. Q. Wu and M. C. Zhou, "Modeling and deadlock control of automated guided vehicle systems," *IEEE/ASME Trans. Mechatron.*, vol. 9, no. 1, pp. 50–57, Mar. 2004.

[246] N. Q. Wu and M. C. Zhou, "Modeling and deadlock avoidance of automated manufacturing systems with multiple automated guided vehicles," *IEEE Trans. Syst., Man, Cybern. B: Cybern.*, vol. 35, no. 6, pp. 1193–1202, Dec. 2005.

[247] N. Q. Wu and M. C. Zhou, "Shortest routing of bi-directional automated guided vehicles avoiding deadlock and blocking," *IEEE/ASME Trans. Mechatron.*, vol. 12, no. 1, pp. 63–72, Feb. 2007.

[248] N. Q. Wu and M. C. Zhou, "Deadlock resolution in automated manufacturing systems with robots," *IEEE Trans. Autom. Sci. Eng.*, vol. 4, no. 3, pp. 474–480, Jul. 2007.

[249] N. Q. Wu, M. C. Zhou, and G. Hu, "On Petri net modeling of automated manufacturing systems," in *Proc. IEEE Int. Conf. Netw., Sens.. Control*, London, U.K., 2007, pp. 228–233.

[250] N. Q. Wu, M. C. Zhou, and Z. W. Li, "Resource-oriented Petri nets for deadlock avoidance in flexible assembly systems," *IEEE Trans. Syst., Man, Cybern. A: Syst. Humans*, vol. 38, no. 1, pp. 56–69, Jan. 2008.

[251] N. Q. Wu and M. C. Zhou, "Process vs resource-oriented Petri net modeling of automated manufacturing systems," *Asian J. Control*, vol. 12, no. 3, pp. 267–280, 2010.

[252] N. Q. Wu and M. C. Zhou, “Petri net modeling and one-step look-ahead maximally permissive deadlock control of AMS,” *ACM Trans. Embedded Comput. Syst.*, 2011, to be published.
[253] N. Q. Wu, F. Chu, C. B. Chu, and M. C. Zhou, “Hybrid Petri net modeling and schedulability analysis of high fusion point oil transportation under tank grouping strategy for crude oil operations in refinery,” *IEEE Trans. Syst., Man Cybern. C: Appl. Rev.*, vol. 40, no. 2, pp. 159–75, Mar. 2010.
[254] N. Q. Wu, F. Chu, C. B. Chu, and M. C. Zhou, “Short-term schedulability analysis of multiple distiller crude oil operations in refinery with oil residency time constraint,” *IEEE Trans. Syst., Man Cybern. C: Appl. Rev.*, vol. 39, no. 1, pp. 1–16, Jan. 2009.
[255] N. Q. Wu and M. C. Zhou, *System Modeling and Control with Resource-Oriented Petri Nets*. New York: CRC Press, 2010.
[256] R. A. Wysk, N. S. Yang, and S. Joshi, “Detection of deadlocks in flexible manufacturing cells,” *IEEE Trans. Robot. Autom.*, vol. 7, no. 6, pp. 853–859, Dec. 1991.
[257] R. A. Wysk, N. S. Yang, and S. Joshi, “Resolution of deadlocks in flexible manufacturing systems: Avoidance and recovery approaches,” *J. Manuf. Syst.*, vol. 13, no. 2, pp. 128–138, 1994.
[258] X. L. Xie and M. D. Jeng, “ERCN-merged nets and their analysis using siphons,” *IEEE Trans. Robot. Autom.*, vol. 15, no. 4, pp. 692–703, Aug. 1999.
[259] K. Y. Xing, B. S. Hu, and H. X. Chen, “Deadlock avoidance policy for Petri-net modelling of flexible manufacturing systems with shared resources,” *IEEE Trans. Automat. Control*, vol. 41, no. 2, pp. 289–295, Feb. 1996.
[260] 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 Trans. Syst., Man, Cybern. A: Syst. Humans*, vol. 39, no. 1, pp. 188–199, Jan. 2009.
[261] 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 Trans. Syst., Man, Cybern. A: Syst. Humans*, vol. 41, no. 1, pp. 74–84, Jan. 2011.
[262] A. Yalcin and T. O. Boucher, “Deadlock avoidance in flexible manufacturing systems using finite automata,” *IEEE Trans. Robot. Autom.*, vol. 16, no. 4, pp. 424–429, Aug. 2000.
[263] E. Yamalidou and J. Kantor, “Modeling an optimal control of discrete-event chemical processes using Petri nets,” *Comput. Chem. Eng.*, vol. 15, no. 7, pp. 503–519, 1991.
[264] E. Yamalidou, J. O. Moody, and P. J. Antsaklis, “Feedback control of Petri nets based on place invariants,” *Automatica*, vol. 32, no. 1, pp. 15–28, 1996.
[265] M. Yamauchi and T. Watanabe, “Algorithms for extracting minimal siphons containing specified places in a general Petri net,” *IEICE Trans. Fund.*, vol. 82, no. 11, pp. 566–575, 1999.
[266] M. M. Yan, Z. W. Li, N. Wei, and M. Zhao, “A deadlock prevention policy for a class of Petri nets S^3PMR,” *J. Inf. Sci. Eng.*, vol. 25, no. 1, pp. 167–183, 2009.
[267] W. C. Yeh, “Real-time deadlock detection and recovery for automated manufacturing systems,” *Int. J. Adv. Manuf. Technol.*, vol. 20, pp. 780–786, 2002.
[268] Z. M. Zhang and W. M. Wu, “Design of Petri net-based non-redundant and maximally permissive liveness-enforcing supervisors for flexible manufacturing systems,” *ACM Trans. Embedded Comput. Syst.*, 2011, to be published.
[269] C. F. Zhong and Z. W. Li, “A deadlock prevention approach for flexible manufacturing systems without complete siphon enumeration of their Petri net models,” *Eng. Comput.*, vol. 25, no. 3, pp. 269–278, 2009.
[270] C. F. Zhong and Z. W. Li, “On self-liveness of a class of Petri net models for flexible manufacturing systems,” *IET Control Theory Appl.*, vol. 4, no. 3, pp. 403–410, 2010.
[271] M. C. Zhou and F. DiCeasre, “Parallel and sequential exclusions for Petri net modeling for manufacturing systems,” *IEEE Trans. Robot. Autom.*, vol. 7, no. 4, pp. 515–527, 1991.
[272] M. C. Zhou and F. DiCesare, “A hybrid methodology for synthesis of Petri nets for manufacturing systems,” *IEEE Trans. Robot. Autom.*, vol. 8, no. 3, pp. 350–361, Jun. 1992.
[273] M. C. Zhou, F. DiCesare, and D. Rudolph, “Design and implementation of a Petri net supervisor for a flexible manufacturing system,” *Automatica*, vol. 28, no. 6, pp. 1199–1208, 1992.
[274] M. C. Zhou and F. DiCesare, *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*. Boston, MA: Kluwer, 1993.
[275] M. C. Zhou (Ed.), *Petri Nets in Flexible and Agile Automation*, Norwell, MA: Kluwer, 1995.
[276] M. C. Zhou and K. Venkatesh, *Modelling, Simulation and Control of Flexible Manufacturing Systems: A Petri Net Approach*. Singapore: World Scientific, 1998.
[277] M. C. Zhou and M. P. Fanti, “Deadlock Resolution in Computer-Integrated Systems,” *IEEE Trans. Syst., Man, Cybern. A: Syst. Humans*, vol. 34, no. 1, 2004 (special issue).
[278] M. C. Zhou and M. P. Fanti (Eds)., *Deadlock Resolution in Computer-Integrated Systems*. New York: Marcel-Dekker, 2005.
[279] M. C. Zhou and Z. W. Li, “Guest editorial: Petri nets and agile manufacturing,” *Trans. Inst. Meas. Control*, vol. 33, no. 1, pp. 3–8, Feb. 2011.
[280] M. C. Zhou and Z. W. Li, “Petri Nets for System Control and Automation,” *Asian J. Control*, vol. 12, no. 3, pp. 237–239, 2010, (special issue).
[281] B. Zouari and K. Barkaoui, “Parameterized supervisor synthesis for a modular class of discrete event systems,” in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Washington, DC, 2003, pp. 1874–1879.
[282] R. Zurawski and M. C. Zhou, “Petri Nets in Manufacturing,” *IEEE Trans. Ind. Electron.*, vol. 41, no. 6, 1994 (special issue).
[283] R. Zurawski and M. C. Zhou, "Petri Nets and Industrial Applications: A Tutorial," *IEEE Trans. Ind. Electr.*, vol. 41, no. 6, pp. 567–583, Dec. 1994.

ZhiWu Li (M’06–SM’07) received the B.S., M.S., and Ph.D. degrees in mechanical engineering, automatic control, and manufacturing engineering, respectively, from Xidian University, Xi’an, China, in 1989, 1992, and 1995, respectively.

He joined Xidian University, in 1992, where he is currently a Professor with the School of Electro-Mechanical Engineering and the Director of Systems Control and Automation Group. From June 2002 to July 2003, he was a Visiting Professor with the Systems Control Group, Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada. From February 2007 to February 2008, he was a Visiting Scientist with the Laboratory for Computer-Aided Design and Lifecycle Engineering, Department of Mechanical Engineering, Technion-Israel Institute of Technology, Technion City, Haifa, Israel. Since November 2008, he has been a Visiting Professor with Automation Technology Laboratory, Institute of Computer Science, Martin-Luther University of Halle-Wittenburg, Halle (Saale), Germany. He was a Senior Visiting Scientist with Conservatoire National des Arts et Métiers, Paris, France, supported by the program Research in Paris in 2010. Since December 2010, he has been a Visiting Professor with King Saud University, Riyadh, Saudi Arabia, supported by the Visiting Professor Program launched by King Saud University. He serves as a Host Professor of Research Fellowship for International Young Scientists, National Natural Science of Foundation of China. He is the author or coauthor of more than 120 publications including two book chapters in *Deadlock Resolution in Computer-Integrated Systems* (New York: Marcel Dekker, 2005) and in *Reconfigurable Embedded Control Systems: Applications for Flexibility and Agility* (Hershey, PA: IGI Global Press, 2011). He is a coauthor with M. Zhou, *Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach*, (Berlin, Germany: Springer, 2009) and *Modeling, Analysis and Deadlock Control in Automated Manufacturing Systems* (Beijing, China: Science Press, 2009, in Chinese). His current research interests include Petri net theory and application, supervisory control of discrete event systems, workflow modeling and analysis, and systems integration.

Dr. Li is the General Co-Chair of the IEEE International Conference on Automation Science and Engineering, Washington, DC, August 23–26, 2008. He is a financial Co-Chair of the IEEE International Conference on Networking, Sensing, and Control, March 26–29, 2009, a member of International Advisory Committee, 10th International Conference on Automation Technology, June 27–29, 2009, a Co-Chair of the program committee of the IEEE International Conference on Mechatronics and Automation, August 24–27, 2010, and a member of the program committees of many international conferences. He serves an

Associate Editor of the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, PART A: SYSTEMS AND HUMAN BEINGS, *International Journal of Embedded Control Systems, IST Transactions of Robotics, Automation & Mechatronics—Theory & Applications*, and *IST Transactions of Control Engineering—Theory and Applications*. He is a Guest Editor of Special Issue on "Petri Nets for System Control Automation" in the *Asian Journal of Control*, Special Issue on "Petri Nets and Agile Manufacturing" in the *Transactions of the Institute of Measurement and Control*, and Special Issue on Modeling and Verification of Discrete Event Systems in the *ACM Transactions on Embedded Computing Systems*. He is a member of Discrete Event Systems Technical Committee of the IEEE Systems, Man, and Cybernetics Society. He serves as a frequent reviewer for more than 20 international journals including a number of the IEEE Transactions as well as many international conferences. He is listed in Marquis Who's Who in the world, 27th Edition, 2010. He is a recipient of Alexander von Humboldt Research Grant, Alexander von Humboldt Foundation, Germany. He is the Founding Chair of Xi'an Chapter of the IEEE Systems, Man, and Cybernetics Society.

NaiQi Wu (M'04–SM'05) received the M.S. and Ph.D. degrees in systems engineering from Xi'an Jiaotong University, Xi'an, China, in 1985 and 1988, respectively.

From 1988 to 1995, he was with the Chinese Academy of Sciences, Shenyang Institute of Automation, Shenyang, China, and from 1995 to 1998, with Shantou University, Shantou, China. From 1991 to 1992, he was a Visiting Scholar with the School of Industrial Engineering, Purdue University, West Lafayette, IN. He was a Visiting Professor with the Department of Industrial Engineering, Arizona State University, Tempe, in 1999, with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, in 2004, with the Industrial Systems Engineering Department, Industrial Systems Optimization Laboratory, University of Technology of Troyes, Troyes, France, from 2007 to 2009, and with the Laboratoire IBISC EA 4526, Université d'Evry Val d'Essonne, respectively. He is currently a Professor of industrial and systems engineering with the Department of Industrial Engineering, School of Mechatronics Engineering, Guangdong University of Technology, Guangzhou, China. His research interests include production planning and scheduling, manufacturing system modeling and control, discrete event systems, Petri net theory and applications, logistics and transportation, and information assurance. He is the author or coauthor of many papers published in *International Journal of Production Research*, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, IEEE/ASME TRANSACTIONS ON MECHATRONICS, IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING, *Journal of Intelligent Manufacturing, Production Planning and Control*, *Computers & Chemical Engineering*, and *Robotics and Computer Integrated Manufacturing*.

Dr. Wu is an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, PART C and IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, and the Editor-in-Chief of the *Industrial Engineering Journal*. He is Program Co-Chair of *2011 IEEE International Conference on Networking, Sensing and Control*, Beijing, China, April 2012. He was a Program Committee Member of the 2003–2010 IEEE International Conference on Systems, Man, and Cybernetics, a Program Committee Member of the 2005–2010 IEEE International Conference on Networking, Sensing and Control, a Program Committee Member of the 2006–2010 IEEE International Conference on Automation Science and Engineering, a Program Committee Member of the 2006 IEEE International Conference on service systems and service management, a Program Committee Member of the 2007 International Conference on Engineering and Systems Management, and reviewer for many international journals.

MengChu Zhou (S'88–M'90–SM'93–F'03) received the B.S. degree in electrical engineering from Nanjing University of Science and Technology, Nanjing, China, in 1983, the M.S. degree in automatic control from the Beijing Institute of Technology, Beijing, China, in 1986, and the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, in 1990.

He joined the New Jersey Institute of Technology (NJIT), Newark, in 1990, where he is currently a Professor of electrical and computer engineering and the Director with Discrete-Event Systems Laboratory. He is also a Professor of The Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 200092, China. He has more than 390 publications including nine books, 130+ journal papers, and 17 book chapters. He coauthored (with F. DiCesare) *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems* (Boston, MA: Kluwer, 1993), edited *Petri Nets in Flexible and Agile Automation* (Boston, MA: Kluwer, 1995), coauthored (with K. Venkatesh) *Modeling, Simulation, and Control of Flexible Manufacturing Systems: A Petri Net Approach* (Singapore: World Scientific, 1998), coedited (with M. P. Fanti) *Deadlock Resolution in Computer-Integrated Systems* (New York: Marcel Dekker, 2005), coauthored (with H. Zhu) *Object-Oriented Programming in C++: A Project-Based Approach* (Tsinghua, China: Tsinghua Univ. Press, 2005), coauthored (with B. Hruz) *Modeling and Control of Discrete Event Dynamic Systems* (New York: Springer, 2007) and coauthored (with Z. Li) *Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach* (New York: Springer, 2009). In the Scopus database, in the area of Petri nets, he ranks as the top one in terms of the number of publications and top two in terms of the number of citations. He ranks the top one in the area of automated manufacturing systems in both indices. His research interests include Petri nets, automated manufacturing systems, wireless *ad hoc* and sensor networks, system security, semiconductor manufacturing, life-cycle engineering product design, and embedded control.

Dr. Zhou was the recipient of NSF's Research Initiation Award, CIM University- LEAD Award by the Society of Manufacturing Engineers, Perlis Research Award by NJIT, Humboldt Research Award for U.S. Senior Scientists, Leadership Award and Academic Achievement Award by the Chinese Association for Science and Technology-USA, the Asian American Achievement Award by the Asian American Heritage Council of New Jersey, and the Distinguished Lecturership of the IEEE System, Man and Cybernetics (SMC) Society. He has been invited to lecture in Australia, Canada, China, France, Germany, Hong Kong, Italy, Japan, Korea, Mexico, Taiwan, and the U.S., and served as a plenary speaker for several conferences. He served as Associate Editor of the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION from 1997 to 2000, and the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING from 2004 to 2007, and is currently the Editor of the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: PART A, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and the Editor-in-Chief of the *International Journal of Intelligent Control and Systems*. He has served as a Guest Editor for many journals including the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS and the IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING. He was the General Chair of the IEEE Conference on Automation Science and Engineering, Washington, DC, August 23–26, 2008, the General Co-Chair of 2003 IEEE International Conference on SMC, Washington, DC, October 5–8, 2003, the Founding General Co-Chair of 2004 IEEE International Conference on Networking, Sensing and Control, Taipei, Taiwan, March 21–23, 2004, and the General Chair of 2006 IEEE International Conference on Networking, Sensing and Control, Ft. Lauderdale, FL, April 23–25, 2006. He was the Program Chair of 2010 IEEE International Conference on Mechatronics and Automation, 2005 IEEE International Conference on Networking, Sensing and Control, 1998 and 2001 IEEE International Conference on SMC and 1997 IEEE International Conference on Emerging Technologies and Factory Automation. He also served other chair positions including the Local Arrangement Chair of 2007 American Control Conference. He organized and chaired over 80 technical sessions and served on program committees for many conferences. He has led or participated in 40+ research and education projects with total budget over \$10 M, funded by the National Science Foundation, Department of Defense, Engineering Foundation, New Jersey Science and Technology Commission, and industry. He was the Founding Chair of Discrete Event Systems Technical Committee of IEEE SMC Society, and the Founding Chair of Semiconductor Manufacturing Automation Technical Committee of IEEE Robotics and Automation Society. He is a Life Member of Chinese Association for Science and Technology and served as its President in 1999.