Information Sciences 363 (2016) 178–189

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A novel method for deadlock prevention of AMS by using resource-oriented Petri nets

Hefeng Chen[b], NaiQi Wu[a,b,*], MengChu Zhou[a,c]

[a] Macau Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau
[b] Department of Industrial Engineering, School of Electro-Mechanical Engineering, Guangdong University of Technology, Guangzhou 510006, China
[c] Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102-1982, USA

ARTICLE INFO

Article history:
Received 3 December 2014
Revised 12 May 2015
Accepted 10 August 2015
Available online 18 August 2015

Keywords:
Deadlock prevention
Automated manufacturing system
Petri net

ABSTRACT

Based on the systems of simple sequential processes with resources (S^3PR) model, the existing methods involve prohibitive computation to synthesize a deadlock prevention controller for automated manufacturing systems (AMS). To reduce the computation, this work studies this problem by using a resource-oriented Petri net (ROPN) model. By revealing the relationship between the bad markings and structural properties of an ROPN, it presents a method such that a deadlock prevention controller can be obtained by simple calculation. By such a controller, for each strongly connected subnet in an ROPN, only one control place is needed such that it is structurally very simple. Furthermore, a condition is given under which a maximally permissive controller can be efficiently obtained, which was never seen before. Examples are used to show the application and performance of the proposed method.

1. Introduction

In an automated manufacturing system (AMS), different types of products are processed concurrently. These products are processed according to their pre-specified routes. They compete for a set of resources such as machines, buffers, and tools [20]. In this situation, circular waiting may occur, leading to a deadlock. When a deadlock occurs, some jobs in the system can never be finished or the entire system enters a standstill state. Thus, a deadlock is a highly undesired state and should be completely avoided such that full automation can be realized. Deadlocks in an AMS can be avoided by properly allocating the resources in the system to the processes. This can be done by using control policies that restrict some resource allocations. In the past more than two decades, great attention was paid to deadlock resolution in AMSs by using different control policies [11,59,66].

In general, three strategies are identified for deadlock resolution in AMSs [9]: (1) deadlock prevention [2,9,11,19,20,22–25,31,33,50] that synthesizes a supervisory controller for an AMS such that the resulting system is forced to be deadlock-free; (2) a detection and recovery strategy that allows deadlocks to occur and then technique and policy are used to detect and recover from them [17,44]; and (3) deadlock avoidance that uses on-line control policies to dynamically allocate resources in the system based on feedback information on the current resource allocation such that the system is deadlock-free [1,10,12,14,16,27,30,33–39,42,43,49,51]. When performing deadlock resolution for AMSs, one faces two critical issues, i.e., permissiveness and computational complexity. A control policy is said to be maximally permissive, if any state that is not deadlock or does not necessarily

---

[*] Corresponding author at: Tel.: +853 8897 1991.
E-mail address: nqwu@must.edu.mo, naiqiwu@gmail.com (N. Wu).

http://dx.doi.org/10.1016/j.ins.2015.08.016

lead to a deadlock can be allowed to occur. To make controller design realizable, it is desired that a design algorithm is of polynomial complexity. By deadlock prevention, the supervisory controller is synthesized off-line and the system can be automatically realized when it is put into operation. This results in a low on-line computational burden. It is this characteristic that makes the deadlock prevention strategy more attractive than the others.

In view of the models for deadlock resolution for AMSs, a Petri net is the main tool since it can describe concurrency and asynchrony well [9,14,20,32,34–39,45,47]. For deadlock prevention, a supervisory controller can be synthesized based on Petri net models called systems of simple sequential processes with resources (S^3PR) or their extended ones [9,20,22,23,28,31,32,45]. According to Wu and Zhou [40,41], they belong to process-oriented Petri net (POPN) models pioneered by Zhou et al. [52–58]. If a Petri net is made to be live, it must be deadlock-free. One way to make an S^3PR live is to control its siphons such that every siphon can never be emptied. This requires one to identify the siphons and control them by adding some control places and arcs such that the resulting model is deadlock-free. In synthesizing a supervisory controller for deadlock prevention, the most concerned issues are: 1) permissiveness; 2) the number of control places to be added or structural complexity of the controller; and 3) computational complexity for finding the controller.

The deadlock prevention based on an S^3PR model is pioneered by Ezpeleta et al. [9]. However, the supervisory controller obtained in [9] is conservative and there are redundant control places. Furthermore, it needs to identify all the siphons and there are induced siphons when a control place is added. Hence, it is very time-consuming to find a controller. To improve the method presented in [9], the concept of elementary siphons is defined for an S^3PR in [20]. To make an S^3PR deadlock-free, one needs to control the elementary siphons only but not all the siphons. Based on this finding, techniques are presented to synthesize a supervisor with a minimal number of control places and, at the same time, the computational efficiency is improved [18,20–24]. Generally, by the above methods, a maximally permissive supervisor cannot be obtained and the computational complexity for finding a supervisor is in theory exponential with the size of the model despite the significant reduction of computational time than other reachability-based method [3–7,31].

Notice that the states that are reachable for a Petri net can be obtained by using reachability tree analysis. Based on information obtained by the reachability graph method, a maximal supervisory controller can be found for S^3PR by using the theory of regions [3–7,13,15,24,25,28,31,32]. Uzam [31] divides the reachable states into deadlock zone (DZ) and deadlock-free zone (DFZ). Based on this state separation, he presents a method to find a maximally permissive (or optimal) controller. However, the obtained controller is not structurally optimal, for there are redundant control places. Aiming at solving the problem of redundant control places, for the reachable states, Ghaffari et al. [13] define forbidden markings, dangerous markings, legal markings, and the set of markings/transition-separation instances (MTSI). With the definition of such markings, another method is proposed to find an optimal controller. By this method, indeed the number of redundant control places is reduced. Nevertheless, this method cannot entirely eliminate redundant control places, especially for large-size AMSs. This problem is further studied in [4,5] and a structurally minimal controller can be obtained. Besides the redundant control place issue, to obtain a controller by reachability analysis and the theory of regions, prohibitive computation is involved. Huang and Pan [15] show that the computational burden can be reduced by using a so-called reduction technology. By combining a siphon control method and the theory of regions, the number of MTSIs is significantly reduced in [24]. In this way, the computational efficiency is improved. Uzam and Zhou [32] propose an iterative approach to find a supervisory controller such that less computation is required although it requires the repeated generation of reachability trees. Piroddi et al. [28] propose a method to select siphons to control and critical markings from reachability analysis. Then, based on the selected siphons and markings, a maximally permissive controller can be obtained iteratively, thereby reducing the computation. Since it is an iterative algorithm, it also requires the repeated generation of reachability trees. It should be pointed out that although the computational efficiency is improved to some extent, the computational complexity is exponential with the size of the model for all the aforementioned methods that use reachability analysis and theory of regions since essentially they need to solve both reachability analysis and integer programming problems. Also, up to now, we do not know that under what condition a maximally permissive deadlock prevention controller can be found by a polynomial algorithm. Thus, it is meaningful to search for efficient techniques to answer these questions.

In studying the deadlock avoidance problem in AMSs, a resource-oriented Petri net (ROPN) model is developed to describe the resource allocation behavior of an AMS in [34–39]. It is a type of colored Petri nets and there is a one-to-one mapping between the places in the model and the resources in an AMS [11,40,41,46]. Thus, an ROPN is very compact and its size is linear with the size of the system. Furthermore, in an ROPN, a deadlock can occur in a circuit only and the number of circuits is limited. Also, it is much easier to identify circuits in an ROPN than siphons in a POPN. With ROPN models, Wu et al. [42] present necessary and sufficient conditions under which maximally permissive policy can be obtained by using a polynomial algorithm. This paper uses such an ROPN model and the theory of regions to synthesize a controller for deadlock prevention. By taking the advantage of ROPN, the relationship between bad markings and structural properties in an ROPN is revealed. Based on this relationship, an efficient deadlock prevention method is proposed. By this method, besides the reachability analysis, a controller is found by only simple calculation and excellent performance is obtained in terms of permissiveness. For such a controller, only one control place is needed for each strongly connected subnet in an ROPN and there is no redundant control place at all. Thus, it is structurally simple. Furthermore, we establish a condition under which a maximally permissive controller can be efficiently obtained. This represents important and new advancement in this field.

The rest of this paper is organized as follows. Section 2 briefly introduces the ROPN modeling method and its properties. Section 3 develops the deadlock prevention method. Examples are used to show the applications and performance of the proposed method in Section 4. Section 5 concludes the work.