Computers & Industrial Engineering 72 (2014) 297–305

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Deadlock-free scheduling for flexible manufacturing systems using Petri nets and heuristic search ☆

Hang Lei [a], Keyi Xing [a,*], Libin Han [a], Fuli Xiong [a], Zhaoqiang Ge [b]

[a] The State Key Laboratory for Manufacturing Systems Engineering and the Systems Engineering Institute, Xi'an Jiaotong University, Xi'an 710049, PR China
[b] Department of Applied Mathematics, Xi'an Jiaotong University, Xi'an 710049, PR China

ARTICLE INFO

Article history:
Received 29 August 2013
Received in revised form 1 April 2014
Accepted 6 April 2014
Available online 18 April 2014

Keywords:
Scheduling
Petri net
Flexible manufacturing system
Deadlock control policy
Heuristic search

ABSTRACT

Deadlock-free control and scheduling are two different problems for flexible manufacturing systems (FMSs). They are significant for improving the behaviors of the systems. Based on the Petri net models of FMSs, this paper embeds deadlock control policies into heuristic search algorithm, and proposes a deadlock-free scheduling algorithm to minimize makespan for FMSs. Scheduling is performed as heuristic search in the reachability graph of the Petri net. The searching process is guided by a heuristic function based on firing count vectors of state equation for the Petri net. By using the one-step look-ahead method in the optimal deadlock control policy, the safety of a state is checked. Experimental results are provided to show effectiveness of the proposed heuristic search approach in deadlock-free scheduling for FMSs.

1. Introduction

Flexible manufacturing systems (FMSs) are modern production facilities that are highly adaptable to different production plans. An FMS includes the material transportation system, the buffer, the work-piece warehouse and other resources besides the processing equipments. In an FMS, various parts are concurrently processed and have to share common resources, and then deadlocks may occur during the processing, which are undesirable phenomena. In deadlock situations, the whole system or a part of it remains indefinitely blocked and cannot terminate its task. Therefore, it is highly important to develop efficient control and scheduling algorithms to optimize the system performance while preventing deadlock situations.

The deadlock has been extensively studied from the control viewpoint, and many deadlock control methods have been proposed (Abdallah & Elmaraghy, 1998; Chu & Xie, 1997; Ezpeleta, Colom, & Martinez, 1995; Fanti, Maione, & Turchiano, 2001; Fanti & Zhou, 2004; Huang, Jeng, Xie, & Chung, 2001; Piroddi, Cordone, & Fumagalli, 2008; Reveliotis & Ferreira, 1996; Xing, Hu, & Chen, 1996; Xing, Zhou, Liu, & Tian, 2009; Xing, Zhou, Wang, Liu, & Tian, 2011). They can guarantee the deadlock-free operation of FMS, but do not take operating time into account. Only a few studies address the scheduling problem of deadlock-prone FMSs. Such a problem involves not only the optimization of certain objective function but also the handling of deadlock problems, and therefore is more complex. Ramaswamy and Joshi (1996) provided a mathematical model for a deadlock-free scheduling problem of FMSs with material handing devices and limited buffers, a Lagrangian relaxation heuristic algorithm was used in this paper to simplify the models to search for the optimized average flow time. Jeng and Chen (1998) developed a heuristic algorithm based on the best-first search technique for scheduling FMSs. Abdallah, Elmaraghy, and Elmekkawy (2002) used timed Petri nets (PNs) to model FMSs and proposed a scheduling algorithm. The algorithm generates a partial reachability graph to find the optimal or near-optimal deadlock-free schedule in terms of the firing sequence of transitions in the PN model. Xu and Wu (2004) developed a genetic algorithm based on PN with infinite buffers. They analyzed the deadlock that might occur in the obtained scheduling and added some necessary buffers to avoid the deadlock. Golmakani, Mills, and Benhabib (2006) proposed an automata-based approach to minimize the makespan for deadlock-free scheduling of FMSs. Dashora, Kumar, Tiwari, and Newman (2007) applied extended colored timed PN to model the dynamic behavior of simple sequential processes with resources. A deadlock-free schedule with minimized makespan based on an evolutionary endosymbiotic learning automata algorithm was presented. Wu and Zhou (2007) studied a real-time deadlock-free scheduling problem for semiconductor

☆ This manuscript was processed by Area Editor Manoj Tiwari.
* Corresponding author. Tel.: +86 135 7189 7199.
E-mail addresses: lei.hang.2008@stu.xjtu.edu.cn (H. Lei), kyxing@sei.xjtu.edu.cn (K. Xing), lbhan@sei.xjtu.eud.cn (L. Han), xiongfl1974@163.com (F. Xiong), gezqjd@mail.xjtu.edu.cn (Z. Ge).

http://dx.doi.org/10.1016/j.cie.2014.04.002
0360-8352/

track systems based on colored timed resource-oriented PN. A deadlock avoidance policy (DAP) was used as a lower layer controller. With the support of the DAP, heuristic rules were proposed to schedule the system in real time. Xing, Han, Zhou, and Wang (2012) embedded a deadlock avoidance policy into genetic algorithm to develop a deadlock-free scheduling algorithm for FMSs to minimize the makespan. Based on the deadlock search algorithm (Xing et al., 2009), a one-step look-ahead method was developed to avoid deadlocks by amending the feasibility of chromosomes. A deadlock-free schedule can be obtained.

Branch-and-bound methods (Hariri & Potts, 1997; Lee & Kim, 2004) can obtain optimal solutions of scheduling problems. Their main disadvantage is enumerating large number of nodes. Other search methods such as A∗ search algorithm (Jeng & Chen, 1999; Lee & DiCesare, 1994; Reyes, Yu, & Kelleher, 2002a; Reyes, Yu, Kelleher, & Lloyd, 2002b; Xiong & Zhou, 1998), which is based on branch-and-bound method, attempts to generate and evaluate fewer number of nodes by branching only the most promising nodes at each stage of the search. None of the search methods mentioned above takes into account the deadlock problem. In order to generate deadlock-free schedules, Deadlock detection and prevention mechanism is required to be incorporated into the searching process. This can be done by precisely representing the FMS's discrete-event dynamic behavior to prevent encountering deadlock states. PNs are commonly accepted technique that can explicitly represent the characteristics of FMS.

For an FMS modeled by PN, an optimal schedule can be obtained by generating the reachability graph and finding the optimal path from the initial marking to the final marking based on a given measure of performance. When constructing the reachability graph, deadlock states are designated and the paths to those states are terminated to prevent the system from encountering them. A hybrid heuristic search (Xiong & Zhou, 1998) based on A∗ search algorithm was applied to determining a deadlock-free schedule with respect to the makespan criterion, assuming that each part has only one route to be produced. For PN methods, deadlock recognition is carried out as the schedule is generated and thus, it has to be repeated for every new batch of parts.

Based on the literature, deadlock control and scheduling focus on different sides in their approach. The scheduling emphasizes optimizing the performance of the systems, where the schedule performance is generally defined in terms of the movement of parts through the entire system. In contrast, deadlock control focuses on avoiding or preventing the system from entering into deadlock states. The traditional scheduling of a job shop or a flow shop with the scale $m \times n$, meaning m machines and n parts, assumes that the system has enough buffers or human presence, and considers problems with no deadlock (Pineo, 2000). If the given buffer space is limited, the deadlock freedom of the systems with its obtained schedule cannot be guaranteed. Thus, it is necessary and useful to integrate deadlock control and scheduling together in practice.

In this paper, a new heuristic search method for deadlock-free scheduling of FMS is proposed. This method is described as follows. First, an FMS is modeled as a P-timed PN. A deadlock-free scheduling problem that minimizes the makespan is formulated as searching the reachability graph of the PN. The searching process is guided by a heuristic function based on solution of the state equation, called firing count vector, to predict the total time from the initial marking through the current marking to the final marking. In addition, linear algebraic techniques can be used to obtain an approximate solution of the state equation. Furthermore, this paper uses a one-step look-ahead method for checking the safeness of the next marking in the searching process, and develops a deadlock-free scheduling algorithm. The primary contributions of this paper are the application of the heuristic search based on the state equation for PN to the deadlock-free scheduling problem for the first time and the utilization of DAP in our scheduling problem.

The outline of the paper is organized as follows. Section 2 introduces PN modeling of FMSs for scheduling and reviews deadlock PN controller used in this paper. Section 3 establishes a deadlock-free heuristic scheduling method by imbedding DAP into the heuristic search algorithm. Two examples to show the effectiveness of the proposed scheduling algorithms are given in Section 4. Section 5 gives conclusions.

2. PN models and their DAPs

This section first introduces some definitions and notations of PNs, and then the P-timed PN model of FMS and DAPs for FMS used in this paper.

2.1. Basic PN definitions

A PN is a three-tuple $N = (P,T,F)$, where $P = \{p_1,p_2,\ldots,p_m\}$ is a finite set of places, $T = \{t_1,t_2,\ldots,t_n\}$ is a finite set of transitions with $P \cup T \neq \varnothing$ and $P \cap T = \varnothing$ $F \subseteq (P \times T) \cup (T \times P)$ is the set of directed arcs. Given a net $N = (P,T,F)$, and a node $x \in P \cup T$, 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 $x^\bullet = \{y \in P \cup T | (x,y) \in F\}$. A marking or state of N is M: $P \to \mathbb{Z}^+$, which denotes the number of tokens in each place, where $\mathbb{Z}^+ = \{0,1,2,\Lambda\}$. A PN N with an initial marking M_0 is called a marked PN, denoted as (N,M_0).

A transition $t \in T$ is enabled at a marking M, if $\forall p \in {}^\bullet t$, $M(p) > 0$; this fact will be denoted as $M[t\rangle$. An enabled transition t at M can be fired, resulting in a new reachable marking M', denoted by $M[t\rangle M'$, where $M'(p) = M(p) - 1$, $\forall p \in {}^\bullet t \backslash t^\bullet$, $M'(p) = M(p) + 1$, $\forall p \in t^\bullet \backslash {}^\bullet t$, and otherwise, $M'(p) = M(p)$. A sequence of transitions $\alpha = t_1 t_2 \ldots t_k$ is feasible from a marking M if $M_i[t_i\rangle M_{i+1}$, $i = 1, 2, \ldots, k$, where $M_1 = M$, and M_i, $i = 1, 2, \ldots, k + 1$, are called reachable markings from M. Let $R(N,M)$ denote the set of all reachable markings of N from M.

The incidence matrix $D = [c_{ij}]$ of PN is a matrix D: $P \times T \to \{-1,0,1\}$ such that $c_{ij} = 1$ if $t_j \in {}^\bullet p_i \backslash p_i^\bullet$, $c_{ij} = -1$ if $t_j \in p_i^\bullet \backslash {}^\bullet p_i$, and $c_{ij} = 0$ otherwise. The kth firing vector x_k is an $n \times 1$ column vector of $n - 1$ zeros and one nonzero entry, a value of one in the ith position implies that transition t_i fires at the kth firing. The kth column of the incidence matrix D denotes the change of the marking as the result of firing transition t_k, then $M_k = M_{k-1} + Dx_k$. Suppose that the final marking M_f is reachable from the current marking M through a firing sequence $x_1, x_2, \ldots, x_s$, then the state equation for PN is written as $M_f = M + D(x_1 + x_2 + \cdots + x_s)$. We obtain

$$Du = \Delta M$$

where $\Delta M = M_f - M$, $u = x_1 + x_2 + \cdots + x_s$ is an $n \times 1$ column vector of nonnegative integers and is called the firing count vector.

The composition of two PNs, $N_i = (P_i,T_i,F_i)$, $i \in \{1,2\}$, via the same elements, denoted as $N_1 \otimes N_2 = (P,T,F)$, where $P = P_1 \cup P_2$, $T = T_1 \cup T_2$, and $F = F_1 \cup F_2$.

2.2. P-Timed PN scheduling models of FMSs

An FMS consists of m types of resources and can processes n types of parts. The set of resource type is denoted as $R = \{r_i, i = 1, 2, \ldots, m\}$. The capacity of a resource type r_i is an integer, denoted as $C(r_i)$, indicating the maximum number of parts that such type of resources can simultaneously hold. The set of part type is denoted as $Q = \{q_i, i = 1, 2, \ldots, n\}$. The number of type-$q_i$ parts to be processed is $\varphi(q_i)$. A processing route of a part is a sequence of operations. A part may have more than one route and can choose its route in the processing. Let a type-q part have k processing