Computers & Industrial Engineering 59 (2010) 799–806

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Heuristic search for scheduling flexible manufacturing systems using lower bound reachability matrix ☆

Jonghyun Lee [a,*], Jin S. Lee [b]

[a] Instrumentation and Control Research Group, POSCO Technical Laboratories, 1, Goedong-Dong, Nam-Gu, Pohang, Gyungbuk 790-300, Republic of Korea
[b] Division of Electrical and Computer Engineering, Pohang University of Science and Technology, San 31, Hyoja-Dong, Nam-Gu, Pohang, Gyungbuk 790-784, Republic of Korea

A R T I C L E I N F O

Article history:
Received 28 September 2006
Received in revised form 18 October 2009
Accepted 16 August 2010
Available online 20 August 2010

Keywords:
Petri nets
Flexible manufacturing systems
A^* algorithm
B-net modelling
Heuristic search
Lower bound reachability matrix

A B S T R A C T

For scheduling flexible manufacturing systems efficiently, we propose new heuristic functions for A^* algorithm that is based on the *T*-timed Petri net. In minimizing makespan, the proposed heuristic functions are usually more efficient than the previous functions in the required number of states and computation time. We prove that these heuristic functions are all admissible and one of them is more informed than that using resource cost reachability matrix. We also propose improved versions of these heuristic functions that find a first near-optimal solution faster. In addition, we modify the heuristic function of Yu, Reyes, Cang, and Lloyd (2003b) and propose an admissible version in all states. The experimental results using a random problem generator show that the proposed heuristic functions perform better as we expected.

1. Introduction

Flexible manufacturing system (FMS) is a system that can produce multiple types of products using shared resources such as robots, multipurpose machines, and etc. Its characteristics are described as discrete events, resource sharing, concurrency, routing flexibility, and lot size variety. As the complexity of manufacturing systems increase, the development of efficient scheduling and planning techniques for FMS became an important issue.

Solving a scheduling problem is to determine a sequence of operations in every job so that the makespan is minimized or the utilization of critical machines is maximized while satisfying the manufacturing objectives. But the problem belongs to the class of NP-hard problems for which optimal polynomial algorithms are hard to develop.

To solve the scheduling problem, a number of methods have been proposed in the literature. As a scheduling conflict solution method, the beam search that constructs partial schedules within the beam-depth and evaluates them to choose the best one was proposed by Shih and Sekiguchi (1991). The search is based on simple heuristic rules but does not guarantee global optimization. A linear programming approach was presented for periodic scheduling of systems modeled using Petri nets (PNs) (Onaga, Silva, & Watanebe, 1991). The dispatching rules or conflict resolution rules were used whenever a conflict arised in the systems (Camurri, Franchi, Gandolfo, & Zaccaria, 1993; Huang & Chang, 1992; Takamura & Hatono, 1991). Also, the search problem using a branch and bound (B & B) approach was studied by Chen, Yu, and Zhang (1993) and Lloyd, Yu, and Konstas (1995). Although these studies use heuristic search for PN model, their performances were not good enough to apply to FMS applications.

To find the optimal or a near-optimal solution for FMS scheduling problems, Lee and Dicesare (1994) have used A^* algorithm with several heuristic functions. A modified heuristic function used in the A^* search algorithm was proposed by Jeng, Chen, and Lin (1996) and Jeng and Chen (1998) by using the solution of PN state equation. In their algorithm, the search space was limited by the pruning techniques. As a result, the ability to find a first near-optimal solution was enhanced. A hybrid heuristic search using Best First search and Backtracking search was proposed by Xiong and Zhou (1998) to improve the search efficiency.

Although these studies represent efforts to combine PN with systematic heuristic search based on artificial intelligence (AI), they could not always find the optimal solution. To solve this problem, Yu et al. (2003b) proposed an admissible heuristic function that can find the global optimal solution by using resource cost reachability (RCR) matrix and proved that it is admissible. But when they developed their function, they did not consider the remaining time during the transition in their algorithm and hence their function is not truly admissible. In this paper, we propose

☆ This manuscript was processed by Area Editor Gursel A. Suer.

* Corresponding author. Tel.: +82 54 220 5042.
E-mail address: hanahnee@yahoo.co.kr (J. Lee).

0360-8352/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2010.08.006

new heuristic functions of A^* algorithm that are admissible but also more efficient in terms of the search space and computation time than that of using RCR matrix.

This paper is organized as follows. Description of FMS and its PN model is presented in Section 2. Two admissible heuristic functions are proposed and their improved versions that find a first near-optimal solution faster are presented in Section 3. In Section 4, several experimental results are presented by using the proposed heuristic functions and these results are compared with the previous ones. Finally conclusions and future works are discussed in Section 5.

2. FMS description and its PN model

In this section, we first describe in detail the FMS we deal with in this paper and derive its *T*-timed PN model for FMS scheduling. Based on this PN model, we use A^* algorithm and solve FMS scheduling problem in the next section.

A general FMS scheduling problem may be represented as:

- m resources are available $\{R_1, R_2, \cdots, R_m\}$.
- n jobs are to be processed $\{J_1, J_2, \cdots, J_n\}$.
- Each job J_i has s_i sequences with lot size l_i. The lot size means the number of product to be processed in the job.
- Each sequence S_{ij} has t_{ij} tasks ordered by the product processing procedures.
- Each task T_{ijk} is processed in o_{ijk} number of ways. Each way means an operation that completes the task.
- Each operation O_{ijkl} is composed of a resource set R_{ijkl} which is required to execute the operation and has a processing time of p_{ijkl}.
- The resource set R_{ijkl} is composed of r_{ijkl} number of different resources.

Several reasonable assumptions are made in the above FMS descriptions, the details of which are presented by Yu, Reyes, Cang, and Lloyd (2003a).

In prior works, *P*-timed PN model was widely used mainly because its markings or states are well defined in each firing. As a result, scheduling algorithm can be easily developed in the *P*-timed PN model. But, its main drawback is the large number of involved transitions and places which can exponentially increase the computation time. On the other hand, *T*-timed PN model was not actively used because the markings are not well defined particularly from the beginning of the transition firing to the end. In spite of its drawback, the number of transitions and places required in *T*-timed PN is much smaller than those with *P*-timed PN.

Since it is proved by Murata (1989) that *P*-timed PN and *T*-timed PN are equivalent, we adopt *T*-timed PN in this paper mainly to increase the efficiency in the number of states and computation time. To remedy the uncertainty issue during the firing period, we use the so called unavailable token state. Since the concept of time delay can be associated with the transition, tokens can have two possible states: available and unavailable. If a token is available in a place *P*, then it can be considered as an input token for any transition having *P* as an input place. When an output transition *T* of *P* fires, this token becomes unavailable. The unavailable token in *P* will be transferred to the output places of *T* and become available again after a fixed time delay associated with *T*. Hence, for each token marked unavailable, there exists a remaining time of t_l in order for the token to be available. As time passes by, t_l is decreased until it reaches zero. When t_l reaches zero, the token becomes available.

The definition of the general *T*-timed PN is represented as below.

Definition 1. A general *T*-timed PN is a six-tuple $TPN = \{P, T, I, O, M, d\}$ where:

- $P = \{P_1, P_2, \cdots, P_m\}$ is a finite set of places.
- $T = \{T_1, T_2, \cdots, T_n\}$ is a finite set of transitions with $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$.
- I: $P \times T \Rightarrow N^+ \cup \{0\}$ is an input incidence function that defines a weight of directed arcs from places to transitions. Note that N^+ is a set of positive integers.
- O: $T \times P \Rightarrow N^+ \cup \{0\}$ is an output incidence function that defines a weight of directed arcs from transitions to places.
- M: $P \Rightarrow N^+ \cup \{0\}$ is a marking that indicates the number of tokens in each places. Note that M_0 is the initial marking and M_G is the goal marking.
- d: $T \Rightarrow R^+ \cup \{0\}$ is a delaying function that associates the time delay with each transition. Note that R^+ is a set of positive real numbers.

Given an FMS, we generate a *T*-timed PN model using the B-net modelling method proposed by Yu et al. (2003a). An illustrative example based on this model is shown below. The example FMS consists of three jobs and is given in Table 1. All jobs have only one sequence and each sequence has three to five tasks. Each task has less than two operations and each operation uses one resource at a time. The lot size in all jobs is preassigned to one and we define this case as a single lot size problem. When one or more lot sizes of each job are greater than one, we call this as a multiple lot size problem. Note that the numbers in parentheses are processing times required to execute operations with the corresponding resources.

Now, we generate a *T*-timed PN model as shown in Fig. 1, where a place for resource with a same name represents the same resource place. The generated *T*-timed PN is composed of 18 places and 21 transitions. If we construct the *P*-timed PN for the given FMS, the generated PN will have 39 places and 42 transitions which are almost twice as many as those of the *T*-timed PN. The lot size is represented by the number of tokens in the initial buffer places, J_0, J_1, J_2. In single lot size problem, the number of tokens in the initial buffer places is all assigned to one as shown in Fig. 1. And, at least one of the initial buffer places will have two or more tokens in multiple lot size problem.

In scheduling such an FMS, we adopt the widely used *L*1 algorithm (Lee & Dicesare, 1994) which is an application of traditional AI formulation of well-known A^* algorithm (Pearl, 1984) to the FMS scheduling problem based on PN. The algorithm is as follows:

1. Put the initial marking M_0 in OPEN.
2. If OPEN is empty, exit with failure.
3. Remove the marking from OPEN and put the marking in CLOSED where the heuristic function $f(M) = g(M) + h(M)$ is the minimum. Note that the function $g(M)$ is the actual cost generated while transferring from the initial marking M_0 to the current marking M and the heuristic function $h(M)$ is

Table 1
A simple FMS example.

FMS	Job0 Sequence0	Job1 Sequence0	Job2 Sequence0
Task0	M1(7) or M2(8)	M1(6) or M2(5)	M1(8) or M3(5)
Task1	M2(4)	M2(4) or M3(2)	M2(2)
Task2	M1(7) or M3(4)	M1(6)	M2(6) or M3(4)
Task3	N/A	M1(3) or M2(2)	M1(4) or M2(2)
Task4	N/A	N/A	M1(2) or M3(3)