Computers & Industrial Engineering 104 (2017) 188–200

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Total energy consumption optimization via genetic algorithm in flexible manufacturing systems

Xiaoling Li, Keyi Xing *, Yunchao Wu, Xinnian Wang, Jianchao Luo

The State Key Laboratory for Manufacturing Systems Engineering, and Systems Engineering Institute, Xi'an Jiaotong University, Xi'an 710049, China

ARTICLE INFO

Article history:
Received 2 February 2016
Received in revised form 28 July 2016
Accepted 9 December 2016
Available online 28 December 2016

Keywords:
Flexible manufacturing system
Genetic algorithm
Petri net
Total energy consumption optimization
Scheduling

ABSTRACT

In recent years, there has been growing interest in reducing energy consumption and emissions of manufacturing systems. Except for adopting new equipment or techniques, scheduling is crucial to reduce the total energy consumption of manufacturing systems. This paper focuses on the scheduling problem for flexible manufacturing systems (FMSs) with the objective of minimizing the total energy consumption, and proposes a novel scheduling algorithm for FMSs based on Petri net models and genetic algorithm. Considering that energy consumptions in different states of resources are different, this paper takes two ways for calculating total energy consumptions. In the proposed genetic algorithm, a potential schedule is represented by a chromosome consisting of route selection and operation sequence. Crossover and mutation operations are performed on the operation sequence to guarantee the population diversity. For deadlock-prone FMSs, not all chromosomes can be directly decoded to a feasible schedule. To check the feasibility of chromosomes and convert infeasible chromosomes into feasible ones, a repair algorithm is developed with the help of the deadlock avoidance policy. Experiment results on a typical FMS and an industrial stamping system are provided to show the effectiveness of our proposed scheduling algorithm.

1. Introduction

Energy is the necessary resource for manufacturing systems which can be in various forms such as oil, gas, and electricity. In the last 50 years the consumption of energy by the industrial sector has more than doubled and industry currently consumes about half of the world's energy (Mouzon, Yildirim, & Twomey, 2007). The increasing energy prices and requirements to reduce emissions pose new challenge for manufacturing enterprises. Therefore, energy consumption study becomes a more important issue, and has drawn increasing attention in recent years (Giret, Damien, & Vittal, 2015).

Adopting advanced equipment and hardware can certainly reduce some energy consumption (Mori, Fujishima, Inamasu, & Oda, 2011), while finding an optimal or near-optimal schedule for manufacturing systems also can be an effective way to save energy (Giret et al., 2015). So far a few researchers have paid attention to the total energy consumption scheduling of manufacturing systems, and mostly on the job-shop and flow-shop systems. Several classes of energy optimization problems are studied for job-shop system (Liu, Dong, Lohse, Petrovic, & Gindy, 2013; Zhang & Chiong, 2015; Zhang, Li, & Gao, 2013), single machine system (Shrouf, Ordieres-Meré, García-Sánchez, & Ortega-Mier, 2014), and flexible flow-shop system (Dai, Tang, Giret, Salido, & Li, 2013; Fang, Uhan, Zhao, & Sutherland, 2011).

FMSs are a special kind of manufacturing systems in which various parts are concurrently processed and have to compete for limited resources. Without appropriate control or scheduling policies, deadlocks may occur, under which the whole system or a part of it remains indefinitely blocked and cannot terminate its task. Hence, the scheduling problem of FMSs becomes more difficult when taking deadlock situation into account. The existed energy scheduling methods for job-shop and flow-shop systems cannot be directly applied to deadlock-prone FMSs.

The scheduling problems of deadlock-prone FMSs mainly concentrate on optimizing makespan and time related objectives (Abdallah, Elmaraghy, & Elmekkawy, 2002; Baruwa, Piera, & Guasch, 2015; Dashora, Kumar, Tiwari, & Newman, 2007; Han, Xing, Chen, Lei, & Wang, 2013; Lei, Xing, Han, Xiong, & Ge, 2014; Luo, Xing, Zhou, Li, & Wang, 2015; Ramaswamy & Joshi, 1996; Xing, Han, Zhou, & Wang, 2012). For energy optimization scheduling problem of deadlock-prone FMSs, to the author's knowledge, the existing research work is much less. Only Pang and Le (2014) studied such a problem. Based on weighted place-timed Petri net

* Corresponding author.
E-mail address: kyxing@mail.xjtu.edu.cn (K. Xing).

http://dx.doi.org/10.1016/j.cie.2016.12.008
0360-8352/

model, they formulated the optimization of the productive and idle energy consumption for deadlock-prone FMSs in terms of mathematical programming, and the deadlock is prevented by setting mathematical constraints to avoid circular waits in the system. A reachability graph-based discrete dynamic programming approach is proposed to generate near energy-optimal schedules. But as we know from (Xing, Zhou, Liu, & Tian, 2009; Xing, Zhou, Wang, Liu, & Tian, 2011), the circular waits in the considered FMSs can be characterized by *maximal perfect resource transition circuits* (MPRT-circuits). The number of MPRT-circuits is increasing exponentially with the system size in the worst case. Thus, setting mathematical constraints for avoiding circular waits is not easy to be implemented. In this paper, the deadlock is addressed with a polynomial deadlock avoidance algorithm.

This paper focuses on a scheduling problem which minimizes the total energy consumption of FMSs. To solve such scheduling problems, a new effective genetic scheduling algorithm is proposed based on the place-timed Petri net model of the systems. According to different energy consumption levels of resources in various states, two ways of calculating total energy consumptions are proposed. One of them divides resource states into two kinds, hold and idle states, while another divides resource states into working, occupied, and idle states. Based on a feasible transition sequence and the earliest firing time of its transitions, the so-called transition and time matrices are constructed. With these two matrices and energy input parameters, two energy consumption functions are established. In our scheduling algorithm, a possible solution is coded as a chromosome which consists of two sections. The first section records the route information of each part, and the second is a sequence of operations of all parts. A chromosome can be decoded uniquely to a transition sequence, but such a transition sequence maybe infeasible, that is, it may lead the system to deadlock. On the other hand, the calculation of energy consumption is only meaningful for feasible transition sequences. Thus it is necessary to check the feasibility of transition sequences or corresponding chromosomes. In this paper, the amending algorithm with polynomial complexity proposed in Xing et al. (2012) is used to check the feasibility of chromosomes and amend infeasible chromosomes into feasible ones. To show the effectiveness of our algorithm in minimizing energy consumption, our scheduling algorithm is tested on some examples.

The rest of this paper is organized as follows. Section 2 introduces FMSs and their Petri net models. The two different resource status modes and their total energy consumption functions are presented in Section 3. Section 4 develops a new deadlock-free genetic algorithm for total energy consumption optimization problem of FMSs. Section 5 presents examples to show the effectiveness of the proposed algorithm. Section 6 concludes the paper.

2. FMSs and their PN models

This section introduces the considered FMSs and their Petri net models for scheduling. The basic definition and notations of Petri nets are in Appendix A and readers can find more details in Murata (1989) and Xing et al. (2009).

An FMS considered in this paper is the same as in Pang and Le (2014) and Xing et al. (2009, 2012). It consists of m types of resources and is able to process n types of parts. The set of resource types is denoted as $R = \{r_i, i \in \mathbb{Z}_m\}$. The capacity of resource type r_i is an integer, denoted as $\chi(r_i)$, indicating the maximum number of parts that such type of resources can simultaneously hold.

The set of part types is denoted as $Q = \{q_i, i \in \mathbb{Z}_n\}$. 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 to be processed on resources; each part may have more than one route. A route of a type-q_i part can be expressed as $\omega_j = o_{is}o_{j1}o_{j2}\ldots o_{jl}o_{ie}$, where o_{is} and o_{ie} are fictitious operations for type-q_i parts, and l is the length of route ω_j.

In our PN model, ω_j is modeled by a path of transitions and places denoted as $\rho(\omega_j) = p_{is}t_{j1}p_{j1}t_{j2}p_{j2}\ldots t_{j(l-1)}p_{j(l-1)}t_{jl}p_{ie}$, where place p_{uv} represents operations o_{uv}. Hence, the marked PN model of processing routes for type-q_i parts can be denoted as

$$(N_i, M_{i_0}) = (P_i \cup \{p_{is}, p_{ie}\}, T_i, F_i, M_{i_0})$$

where P_i is the set of operation places which require resources, M_{i_0} is the initial marking, $M_{i0}(p) = 0$, $\forall p \in P_i$, and $M_{i0}(p_{is}) = \varphi(q_i)$.

In N_i, $\forall t \in T_i$, $|{}^\bullet t| = |t^\bullet| = 1$. If $p \in P_i$ and $|p^\bullet| > 1$, p is called a split place. A part can choose its processing routes at a split place.

For each resource type r_k, we assign a place, called a resource place and denoted also by r_k, for simplicity. Let P_R denote the set of all resource places and $R(p)$ denote the resource required by operation place p. Then, the request and release of resources $R(p)$ can be modeled by adding arcs from $R(p)$ to each transition in ${}^\bullet p$ and from each transition in $p^\bullet$ to $R(p)$. Let $H(r)$ denotes the set of all operation places that require resource r, i.e. $H(r) = \{p \in P | R(p) = r\}$. Let F_R denote the set of arcs related with resource places. Then, the system can be modeled by the following marked PN

$$(N, M_0) = (P \cup P_s \cup P_f \cup P_R, T, F, M_0)$$

where $P = \cup_{i\in\mathbb{Z}_n} P_i$, $P_s = \{p_{is} | i \in \mathbb{Z}_n\}$, $P_f = \{p_{ie} | i \in \mathbb{Z}_n\}$, $T = \cup_{i\in\mathbb{Z}_n} T_i$, $F = F_Q \cup F_R$, $F_Q = \cup_{i\in\mathbb{Z}_n} F_i$. The initial marking M_0 is defined as $M_0(p_{is}) = \varphi(q_i)$, $\forall p_{is} \in P_s$; $M_0(p) = 0$, $\forall p \in P \cup P_f$, and $M_0(r_k) = \chi(r_k)$, $\forall r_k \in P_R$.

Time delay $d(p)$ is assigned to operation place p to denote its processing time. Note that $d(p) = 0$, $\forall p \in P_s \cup P_f \cup P_R$. Such a PN is called as *Petri Net for scheduling* (*PNS*) (Xing et al., 2012).

When all operations of all parts are finished, the system reaches its final marking, denoted as M_f, where $M_f(p) = 0$, $\forall p \in P \cup P_s$; $M_f(p_{ie}) = M_0(p_{is})$, $\forall p_{ie} \in P_f$; and $M_f(r_k) = \chi(r_k)$, $\forall r_k \in P_R$. A sequence of transitions α is called a *feasible* schedule if $M_0[\alpha > M_f$.

Example 1. Consider an FMS that consists of three machines m_1, m_2, and m_3 and two robots r_1 and r_2. The resource set is $R = \{m_1, m_2, m_3, r_1, r_2\}$, $\chi(m_i) = 1$, $i \in \mathbb{Z}_3$, and $\chi(r_i) = 2$, $i \in \mathbb{Z}_2$. The system can process two types of parts, i.e., q_1 and q_2. Type-q_1 part is first moved into machine m_1 or m_3 by robot r_1, then processed on m_1 or m_3, and processed on m_2 after its operation on m_1 or m_3, finally moved by robot r_2 from m_2 after the operation on m_2. Type-q_2 part is first moved into machine m_3 by robot r_2, and then moved by r_1 when the operation on m_3 is finished. Hence, there are two operation sequences for type-q_1 parts: $o_{11}o_{12}o_{13}o_{14}$ and $o_{11}o_{22}o_{23}o_{14}$, the routes are $\omega_1 = o_{1s}o_{11}o_{12}o_{13}o_{14}o_{1e}$ and $\omega_2 = o_{1s}o_{11}o_{22}o_{23}o_{14}\,o_{1e}$, and the paths of transitions and places are $\rho(\omega_1) = p_{1s}t_{11}p_{11}t_{12}p_{12}\,t_{13}p_{13}t_{14}p_{14}t_{15}p_{1e}$ and $\rho(\omega_2) = p_{1s}t_{11}p_{11}t_{22}p_{22}t_{23}p_{23}t_{24}p_{14}t_{15}p_{1e}$, respectively. While the operation sequence for type-q_2 parts is $o_{31}o_{32}o_{33}$, the route is $\omega_3 = o_{2s}o_{31}o_{32}o_{33}o_{2e}$, and the corresponding path of transitions and places is $\rho(\omega_3) = p_{2s}t_{31}p_{31}t_{32}p_{32}t_{33}p_{33}t_{34}p_{2e}$. The model of the system is shown in Fig. 1, where the required processing parts of q_1 and q_2 are 2 and 2, respectively.

Let ${}^{(o)}t$ and $t^{(o)}$ denote the input and output operation places of transition t, respectively, and ${}^{(r)}t$ and $t^{(r)}$ the input and output resource places of t, respectively. For a given marking $M \in R(N, M_0)$, t is operation-enabled at M if $M({}^{(o)}t) > 0$, and t is resource-enabled at M if $M({}^{(r)}t) > 0$. In PNS, only transitions that are both operation and resource-enabled can be fired.

3. Different energy consumptions in FMSs

Energy consumption in FMSs is decided by the states of all equipment or resources, such as machines, robots during the