Computers & Industrial Engineering 66 (2013) 406–417

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Fast reactive scheduling to minimize tardiness penalty and energy cost under power consumption uncertainties ☆

Cao Vinh Le, Chee Khiang Pang *

Department of Electrical and Computer Engineering, National University of Singapore, Singapore

ARTICLE INFO

Article history:
Received 11 March 2013
Received in revised form 12 June 2013
Accepted 9 July 2013
Available online 23 July 2013

Keywords:
Discrete event control
Dynamic scheduling
Energy-efficient manufacturing
Power consumption uncertainties
Reactive scheduling

ABSTRACT

Motivated by the need to deal with uncertainties in energy optimization of flexible manufacturing systems, this paper considers a dynamic scheduling problem which minimizes the sum of energy cost and tardiness penalty under power consumption uncertainties. An integrated control and scheduling framework is proposed including two modules, namely, an augmented discrete event control (ADEC) and a max-throughput-min-energy reactive scheduling model (MTME). The ADEC is in charge of inhibiting jobs which may lead to deadlocks, and sequencing active jobs and resources. The MTME ensures the fulfillment of the innate constraints and decides the local optimal schedule of active jobs and resources. Our proposed framework is applied to an industrial stamping system with power consumption uncertainties formulated using three different probability distributions. The obtained schedules are compared with three dispatching rules and two rescheduling approaches. Our experiment results verify that MTME outperforms three dispatching rules in terms of deviation from Pareto optimality and reduces interrupted time significantly as compared to rescheduling approaches. In addition, ADEC and MTME are programmed using the same matrix language, providing easy implementation for industrial practitioners.

1. Introduction

Flexible manufacturing systems (FMSs) are modern production facilities which possess high flexibility of resource allocation and part routing. A resource is capable of performing multiple jobs, and multiple resources can be used to perform the same job on a part (Abazari, Solimanpur, & Sattari, 2012; Chan, Bhagwat, & Wadhwa, 2008, 2012; Pang, Lewis, Lee, & Dong, 2011). If one monitors the energy consumption of FMS, it is not uncommon to see that different resources require different productive powers and processing times to perform the same job. This variation is due to a multitude of factors, whether predicted or unpredicted, including the resource type, its operating conditions, process parameters, and part type (Abdelaziz, Saidur, & Mekhilef, 2011). To reduce energy cost of FMS, it is crucial to develop effective scheduling algorithms which generate energy-efficient schedules complying with production constraints.

Owing to current looming economic situation and rising energy prices, the reduction of energy cost in FMS has been recently addressed with great efforts in both academia and industry (Du, Chen, Huang, & Yang, 2011; Fang & Lin, 2013). Current research literature on energy-efficient scheduling often deals with the static environments, where power consumption of resources is fixed and no uncertainties would influence job processing after a schedule is executed. Real manufacturing is, however, dynamic and subjected to a wide range of uncertainties. Uncertainties in manufacturing have been classified into two categories, namely, resource-related uncertainties such as machine breakdown, machine degradation, tool wears, and job-related uncertainties such as rush jobs, job cancellation, stochastic processing times (Vieira, Hermann, & Lin, 2003). As such, scheduling under uncertainties, also known as dynamic scheduling, has attracted much attention in recent years (He & Sun, 2012; Horng, Lin, & Yang, 2012; Xiong, Xing, & Chen, 2012). The FMS scheduling problem is non-deterministic polynomial-time hard (NP-hard) in computational complexity theory, but consideration of uncertainties further aggravates its complexity. The existent approaches for dynamic scheduling in FMS can be classified into three categories, namely, the reactive, the proactive, and the predictive–reactive. Each approach has its own pros and cons (Ouelhadj & Petrovic, 2009).

Predictive–reactive scheduling is a scheduling/rescheduling process, in which the baseline schedules are generated offline and the active schedules are revised online in response to real-time uncertainties. The most common predictive-reactive scheduling include completed rescheduling (CR) and partial rescheduling (PR) (Choi & Wang, 2012; Vieira et al., 2003). In theory, CR provides

☆ This manuscript was processed by Area Editor Manoj Tiwari, Ph.D.

* Corresponding author. Address: Department of Electrical and Computer Engineering, National University of Singapore, 4 Engineering Drive 3, Singapore 117583, Singapore. Tel.: +65 6516 7942; fax: +65 6779 1103.

E-mail address: justinpang@nus.edu.sg (C.K. Pang).

0360-8352/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cie.2013.07.006

the optimal schedules, but these schedules are rarely achievable in practice and require prohibitive computational time. In addition, it can result in instability and disruption in manufacturing flows, leading to tremendous production costs. In PR, only jobs and resources affected by the uncertainties are rescheduled. On the other hand, the reactive scheduling is characterized by its capability of real-time decision-making, in which no baseline schedules are generated offline, and decisions are quickly made online using real-time information. Dispatching rules are typical examples of reactive scheduling, in which jobs are selected by sorting them according to some predefined criteria. Dispatching rules are still the most preferred scheduling approaches in industry due to their ease of implementation, low computational cost, and guarantee of schedule stability and feasibility (Ko, Kim, Kim, & Baek, 2010; Mouelhi-Chibani & Pierreval, 2010; Chiang, 2013; Sule, 2007; Tay & Ho, 2008). The main weakness of reactive scheduling is that they cannot globally optimize the overall performance of generated schedules. Proactive scheduling focuses on building a predictive schedule which minimizes the effects of real-time uncertainties (Horng et al., 2012). Baseline schedules are generated offline and will not be revised online. The main difficulty of these approaches is modeling of uncertainties. Computational cost is also an issue, since the stochastic search space is usually huge.

In this paper, a FMS dynamic scheduling problem which minimizes the sum of energy cost and tardiness penalty is considered under power consumption uncertainties. Uncertainties in power consumption are realistic in a dynamic manufacturing environment, as power consumption was verified to be dependent on uncertain factors including machine conditions, tool conditions, and workloads (Abdelaziz et al., 2011). The minimization of energy cost and tardiness penalty is a practical problem which was considered by Fang and Lin (2013) under static environment. Such tradeoff happens when a resource requires shorter time but higher energy to perform a job as compared to others (Fang & Lin, 2013).

To solve the formulated dynamic scheduling problem, this paper proposes a matrix-based integrated control and scheduling framework for a class of FMSs with shared resources and flexible part routing. Such configuration can be encountered in many realistic manufacturing flowlines, job shops, and material handling systems. The proposed framework can be viewed as an aggregation of two interacting modules, an augmented discrete event control (ADEC) and a max-throughput-min-energy reactive scheduling model (MTME). The ADEC has been proposed recently (Le, Pang, Lewis, Gan, & Chan, 2011; Pang, Hudas, Mikulski, Le, & Lewis, submitted for publication), proving to be very efficient in modeling and controlling the large-scale discrete-event dynamics of typical manufacturing systems. In particular, it reduces the model complexity when modeling large-scale FMSs as compared to the traditional conjunctive supervisory tools, such as the discrete event control (DEC) (Bogdan, Lewis, Kovacic, & Mireles, 2006; Pang et al., 2011) and Petri Nets (PNs) (Huang, Shi, & Xu, 2012). The proposed MTME resembles a reactive scheduling approach, which dispatches the imminent jobs and resources quickly and online using real-time power consumption of resources. It includes two 0–1 linear programming submodels, the former maximizes the production throughput and the latter minimizes the energy cost at every dispatching epoch. Both ADEC and MTME are programmed using the same matrix language and function during operational control as a whole, which provide easy implementation for industrial practitioners.

Our proposed framework is tested on an industrial FMS at a stamping company in the Republic of Singapore. The stamping parts are various types of voice coil motor (VCM) yokes used in commercial hard disk drive (HDD) actuators. Power consumption of resources are continually monitored using Rudolf R-DPA96A digital power analyzers (RUDOLFs). RUDOLFs are interfaced with computers via LabVIEW© environment. The schedules obtained by our proposed framework are compared with three dispatching rules, CR, and PR approaches. The experiment results with different batch sizes verify that MTME outperforms the three dispatching rules for all test cases in terms of deviation from Pareto optimality. The PR outperforms MTME when the batch size is small (short schedules), but the reverse is observed when the batch size is larger than 60 parts (long schedules). In terms of mean interrupted time, MTME achieves less than 1 s for all test cases, while the PR and CR cause prohibitive interrupted time (instability) for the FMS.

The rest of the paper is organized as follows. Section 2 describes the FMS and introduces the ADEC model of FMS. Section 3 formulates the dynamic scheduling problem under power consumption uncertainties, while Section 4 provides the formulation of MTME based on the ADEC model. In Section 5, our framework is evaluated based on an industrial stamping system with experiment results and discussions. Finally, our conclusions and future work directions are summarized in Section 6.

2. Background

In an era of intensive competition, manufacturing systems have migrated from conventional fixed-hardware sequential or batch production with dedicated workstations to FMSs with shared resources and flexible part routing. In this section, FMSs are described. It is then proceeded to provide a general description of the ADEC model of FMS, introducing the most significant details and notations (Le et al., 2011; Pang et al., submitted for publication).

2.1. Description of FMS

The FMS class of systems, investigated herein, has the following properties (Bogdan et al., 2006): (a) each part type has a strictly defined sequence of jobs; (b) each job in the system requires one and only one resource; (c) there are choice jobs (jobs which can be performed by alternative resources) and shared resources (resources which can perform different jobs); (d) resource allocation and part routing are flexible; (e) there are no assembly jobs, and (f) jobs are not preemptive, i.e., once assigned, a resource cannot be removed from a job until it is completed.

A FMS consists of a set of resources, denoted by $R = \{r_j, j = 1, 2, \ldots, |R|\}$, to manufacture $|\Pi|$ types of parts, where $|\bullet|$ is a standard term to denote the cardinality of a set. Each resource can be a machine, a buffer, a robotic arm, an automated guided vehicle, and so on. In large-scale FMSs, r_j can denote a pool of similar resources. Resources which can perform multiple jobs are called shared resources, otherwise called nonshared resources.

The set of part types is denoted by $\Pi = \{\pi_q, q = 1, 2, \ldots, |\Pi|\}$, and $\varphi(\pi_q)$ is the number of type-π_q parts (batch size) to be manufactured. Each π_q has a strictly predefined sequence of jobs $\omega_q = v_1^q v_2^q \cdots v_{|\omega_q|}^q$, where v_i^q is the ith job in ω_q and $|\omega_q|$ is the length of ω_q. The set of jobs is denoted by $V = \{v_i^q, q = 1, 2, \ldots, |\Pi|t\}$. In FMSs with flexible part routing, choice jobs are ubiquitous. Therefore, V can be partitioned into two disjoint subsets, $V = V_z \cup V_{nz}$, where V_z and V_{nz} denote the sets of choice and nonchoice jobs, respectively. Let $R(v_i^q)$ be the set of resources which can perform v_i^q. Obviously, $|R(v_i^q)| > 1$ if $v_i^q \in V_z$, and $|R(v_i^q)| = 1$ if $v_i^q \in V_{nz}$. For each π_q, ω_q is associated with two fictitious jobs u^q and y^q called input buffer and output buffer jobs which represent the storage of raw and finished parts, respectively. u^q and y^q do not require any resources, thus $R(u^q) = R(y^q) = \emptyset$.

2.2. ADEC Model of FMS

Let us consider a FMS with part type π_q is characterized a job sequence ω_q properly predefined and a set of available resources