Computers & Operations Research 50 (2014) 128–140

Contents lists available at ScienceDirect

Computers & Operations Research

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

Two-phase branch and bound algorithm for robotic cells rescheduling considering limited disturbance

Pengyu Yan [a], Ada Che [b,*], Xiaoqiang Cai [c], Xiaowo Tang [a]

[a] School of Management and Economics, University of Electronic Science and Technology of China, Chengdu 610054, PR China
[b] School of Management, Northwestern Polytechnical University, Xi'an 710072, PR China
[c] Department of Systems Engineering & Engineering Management, Chinese University of Hong Kong, Hong Kong

ARTICLE INFO

Available online 18 April 2014

Keywords:
Robotic cells rescheduling
Limited disturbance
Branch and bound algorithm
Dynamic enumeration
Search strategy

ABSTRACT

This paper addresses a robotic cell rescheduling problem and focuses on trade-off between the total completion time of all jobs and the disturbance of a reschedule. We first define and measure the disturbance of a reschedule as the deviation of completion time of the jobs already scheduled between the reschedule and the initial schedule. To guarantee the steady performance of the system, we consider a special case that the processing sequence of the jobs already scheduled cannot be changed. The addressed rescheduling problem is transformed into a series of deterministic local scheduling problems with the objective of minimizing the total completion time of all jobs provided that the disturbance is within a given limit. A two-phase branch and bound algorithm is developed to efficiently solve the local scheduling problems. To improve the efficiency of the search procedure, a dynamic enumeration mechanism is applied to eliminate redundant constraints. Furthermore, two search strategies are proposed to direct the search procedure toward finding an optimal solution and a near-optimal solution. Finally, computational results demonstrate the efficiency of our algorithm.

1. Introduction

Robotic cells are increasingly commonplace in automated manufacturing systems, such as printed circuit board (PCB) electroplating lines, semiconductor manufacturing systems, and iron and steel production. In these systems, in order to ensure the production quality, computer-controlled material handling robots are used to transport jobs (parts) from one workstation to another one. Most of scheduling procedures proposed for robotic cells consider deterministic and static production conditions, which is also referred to as the *cyclic robotic cells scheduling* or *cyclic hoist scheduling* in literature [1–9]. However, real robotic cells usually suffer from many unexpected disruptions, such as the arrival of new jobs, workstations or robots breakdown and processing time changes, which have a great impact on the initial schedule and even make it infeasible. Consequently, a reschedule should be found based on the real-time state of the robotic cell when an unexpected disruption happens.

However, the generated reschedule usually deviates much from the initial schedule inevitably, which brings severely impacts on the steady performance of manufacturing systems [10–15]. In robotic cells, the changes in the jobs' completion times affect the subsequent operations in the downstream manufacture and ultimately affect the customers' satisfaction. So, rescheduling in robotic cells requires the trade-off between the initial scheduling objective and the disturbance of the reschedule to the system. On the other hand, this paper assumes that the processing sequence of the jobs already scheduled, referred as *existing* jobs, on each workstation cannot be changed. It is because that the frequent and significant changes in the processing sequence of existing jobs on each workstation may affect the resource re-allocation, create additional setup costs and accordingly change the robot route significantly that may raise the havoc with the robotic cells [10].

The robotic cells scheduling problem addressed in this study differs from the classical flowshop or jobshop scheduling problem in the following aspects. (i) The job transportations in the cell are performed by a material handling robot. As the robot can handle only one job at any time and its transportation time cannot be ignored compared with the jobs' processing times, the productivity of the system is both dependent on the job schedule and the robot schedule [16–19]. (ii) Due to the technical requirements of chemical or electroplating treatments in PCB electroplating and semiconductor manufacturing industries, the job's processing time at each processing stage is required to be within its corresponding time interval defined by lower and upper bounds, which are referred as to *time-window constraints*

* Corresponding author.
E-mail address: ache@nwpu.edu.cn (A. Che).

http://dx.doi.org/10.1016/j.cor.2014.04.002
0305-0548/

[9,20–22]. Any deviations from the time-window will lead to defective jobs. The robotic cells scheduling problem with processing time-window constraints has been proved to be NP-hard in the strong sense [23].

In the past 30 years, deterministic robotic cell scheduling problems have been widely studied. Many researchers have developed diversified scheduling approaches or algorithms, such as mixed integer programming methods [24–27] and branch and bound procedures [9,22,28–32]. Owing to the NP-hard nature of the problem, some researchers turn to a simplified vision of the problem, where the processing times are given constants. As the time-window constraints vanish, several polynomial algorithms have been developed for this simplified problem [6,7,18,33]. For a comprehensive survey on the robotic cells scheduling problems, please refer to [19,34–37].

There have been many studies addressing the rescheduling problems in various manufacturing systems and some of them focused on the impact of the reschedule on the steady performance of the systems. On the rescheduling problems in manufacture systems without material handling robots, Bean et al. [38] developed a match up rescheduling approach for an automobile manufacturer to recover the initial schedule after a period of time. Akturk and Gorgulu [11] proposed a heuristic to determine the match-up time under a machine breakdown and repair the initial schedule in the preceding time interval. Akturk et al. [39] considered parallel machine match-up scheduling with controllable processing time. They proposed exact and heuristic approaches to make a tradeoff between manufacturing cost and match-up time. Other researchers studied the rescheduling problem with limited disturbance to the initial schedule. Wu et al. [40] considered a single machine rescheduling problem and formulate the problem as a bi-criteria model, including the initial scheduling objective and the disturbance of the reschedule. They measure the difference between the start times of operations in the new and initial schedule as the disturbance. Unal et al. [41] considered a single machine rescheduling problem with newly arriving jobs and setup time that depend on job types. They insert the new job into the initial schedule without incurring additional setups or causing jobs to be late. Hall and Potts [12] also considered a single machine rescheduling problem, where a set of new jobs arrive randomly. They treat the disturbances as a cost constraint and a cost objective in their formulations. Recently, Hall and Potts [13] extended their research and considered the case that a set of jobs that have been scheduled become available latter than expected. They proposed a polynomial rescheduling approach to adjust the initial scheduling without causing excessive disturbance.

On the rescheduling problems in the robotic cells, there are few papers closely related to this study. Lamothe et al. [42] first considered the rescheduling problem in the robotic cell with newly arriving jobs. They presented a rescheduling framework and proposed an efficient branch and bound algorithm for the problem. Recently, Zhao et al. [43] considered a robotic cell rescheduling problem with parallel workstations. They formulated the problem as a mixed-integer programming model and used an optimization software package CPLEX to solve it. However, little literature has addressed on decreasing the disturbance of the reschedule in robotic cells. The exception is the work done by Chauvet et al. [10], who considered a special case where any robot or job processing operation already scheduled in the initial schedule cannot be changed in the reschedule. This extremely strict disturbance restriction allowed them to develop a polynomial algorithm by inserting the new job's operations into the idle time intervals in the initial schedule, but it may have a negative impact on the productivity of the system.

In this paper, we address a rescheduling problem in the robotic cell with a newly arriving job and focus on trade-off between the initial scheduling objective and the disturbance to the initial schedule. The contributions of this paper are:

- The disturbance of a reschedule in robotic cells is addressed for the first time, which is defined and measured as the total absolute deviation of the completion times of existing jobs.
- Following the robotic cell rescheduling framework proposed in [42,43], the rescheduling problem is transformed into a serial of deterministic local scheduling problems. Based on the real-time state of the system, we formulate the local problem to minimize the total completion time of existing jobs while controlling the disturbance within a given limit.
- As the processing sequence of the existing jobs cannot be changed, an efficient two-phase branch and bound algorithm is developed to solve the problem. To improve the efficiency of the searching procedure, a dynamic enumeration mechanism [9,42] is applied to eliminate redundant constraints. For large-size problems, a heuristic search strategy is developed to direct the search procedure to efficiently find a near-optimal solution.

This paper is organized as follows. The problem is described in Section 2, where the framework of the robotic cells rescheduling and the disturbance of a reschedule are discussed. In Section 3, the problem is formulated. A two-phase branch and bound algorithm and the computational results are presented in Sections 4 and 5, respectively. We conclude the study and discuss future extensions in Section 6.

2. Problem description

2.1. Problem statement

The robotic cell in this study is composed of $N+2$ workstations and a computer-controlled robot. The workstations are indexed as M_0, M_1, …, M_N, M_{N+1}, where M_0 and M_{N+1} are the input device and output device, respectively. Different jobs (different types of parts) are picked up into the robotic cell and transported from one workstation to another one by the robot. After all the processing stages have been completed, the jobs arrive at the output device. Each transportation of a job is referred to as a *robot move*, which includes: unloading a job from a workstation when the job finishes its current processing stage; transporting the job to another workstation for the next processing stage and loading it on this workstation. Each job visits the workstations following its special processing recipe that describes the job processing route in the cell and its processing time-windows. At any time, the workstations and the robot can handle at most one job. And the *no-wait* constraint is considered, which requires that the job must be transported to the next processing stage as soon as its current processing stage is completed.

We consider the uncertain setting where the jobs arrive at the robotic cell randomly and we assume that each time there is only a single job arriving at the input device. The arriving time and processing recipe of the new job are unknown until it arrives. In order to maintain the high productivity of the cell, a reschedule needs to be generated and meanwhile the disturbance occurred by the reschedule should be kept within a given limit. The disturbance addressed in this study is defined as the total absolute deviation of existing jobs between the initial schedule and the reschedule. Besides, in order to keep the steady performance of the robotic cell, we assume that the processing sequence of existing jobs on each workstation cannot be changed. Consequently, our problem is to reschedule the robotic cell with a newly arriving job to minimize the total completion time of all jobs and control the disturbance under a given limit.

The notation used in this study is listed in Table 1.