Available online at www.sciencedirect.com

SCIENCE DIRECT®

European Journal of Operational Research 161 (2005) 62–72

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/dsw

Resolution of a scheduling problem in a flowshop robotic cell

A. Soukhal *, P. Martineau

Laboratoire d'Informatique, Ecole Polytechnique de l'Université de Tours 64, avenue Jean Portalis, 37200 Tours, France

Available online 27 November 2003

Abstract

We develop in this paper a generic and precise identification of a scheduling problem in a flexible manufacturing system. We consider a flowshop robotic cell that processes several jobs. We assume that there is no intermediate buffer between machines. So, jobs may be blocked when downstream machines are busy. We present an integer programming model to determine the sequence of jobs that minimizes the makespan criterion. In order to solve large size problems, we propose a genetic algorithm (GA). Finally, computational experiments are proposed in order to compare the makespan returned by the GA to a lower bound.

Keywords: Scheduling; Flowshop; Flexible manufacturing cell; Blocking; Genetic algorithm

1. Introduction

In this section, we recall classical hypotheses in the scheduling field, to model workshops. Then, an example points out hiatuses and incoherences that can emerge from the application of scheduling sequences in a real workshop.

1.1. Justifications

Classical hypotheses for schedule computation are numerous and often not enough pertinent for many concrete applications:

- The upstream and downstream stocks of a machine are unlimited. No blockage situation can occur since there is always some free space in stock areas.
- Jobs are immediately transported after being processed on the previous machine. The transporters capacities and transportation times are generally not taken into account.
- Delivery or removal zones are not managed in mutual exclusion. This implies that an infinite number of transporters can be parked in front of the machine without any problem.
- Loading or unloading times are generally not taken into account.

The resolved problem does not really correspond to the actual workshop problem. Thus, it is inadequate for the supervisor to apply a sequence produced by a scheduling algorithm under these

* Corresponding author.
E-mail addresses: soukhal@univ-tours.fr (A. Soukhal), pmartineau@univ-tours.fr (P. Martineau).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2003.08.028

hypotheses. The following examples illustrate this fact.

1.2. Examples

We show why it can be necessary to refine the model of a workshop manufacturing system.

1.2.1. First example: A deadlock situation

Let us consider two machines connected with a single robot. The machines and the robot can process only one job at a time. The intermediate stocks have a zero capacity and the criterion is the makespan. The classical three-field notation of scheduling problems indicates that the problem is a two-machine flowshop if the routings are all identical. Its resolution method searches to fill machines as soon as possible to make them busy and to complete the jobs as soon as possible. Let us suppose that two machines are both loaded and the job on the first machine is completed. The robot takes the job and goes in front of the second machine. The robot grips a job while the second machine processes another one. Because the second machine needs the robot to unload the current job, the absence of a storage area leads to a deadlock.

In this example, we can see that most classical hypotheses in the resolution of scheduling problems are not adapted. Some recent works solve special cases, and take into account these storage places [9].

1.2.2. Second example: A robot may be used as a stock

Let us consider a two-machine flowshop with no intermediate storage area, but with one robot in charge of all transfer between machines. Transfer time is always the same constant value T. This system is classically modeled as a $F2|\text{no-wait}|C_{\max}$ [1] problem. Fig. 1(a) represents a solution to this simplified problem. This sequence may be applied to the real workshop by computing the starting times of all the operations on the machines and on the robot (Fig. 1(b)).

If we identify and if we consider the constraints more precisely, we can have to solve in fact a $F3|\text{stock} = 0, \text{block}|C_{\max}$ problem. A solution to this problem is given in Fig. 1(c). The robot is

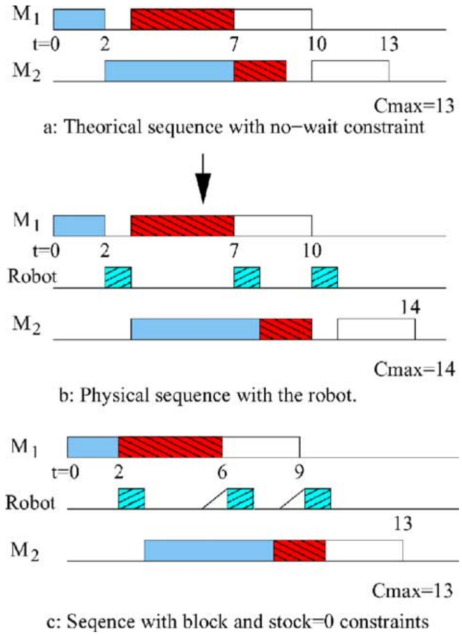

Fig. 1. A flowshop with two machines and a no-wait constraint.

favorably used as an intermediate storage area. This sequence can be transferred to the real workshop. Thus, the resulting makespan has been improved.

The example shows that it is necessary to refine the model, in order to take generally unmodeled constraints into account. Unfortunately, a more complex model leads generally to a more complex scheduling problem.

We propose an original approach composed first, of a precise identification of the problem with a minimum number of simplifying hypotheses and then, of a resolution of a scheduling problem with appropriate methods. We step by step transform a simple and well known model in a more complex. The intermediate models are denoted by using the classical three-fields notation of [6]. Finally, the aim is to get a direct translation of the scheduling algorithm results into control orders [4].

In Section 2, the basic idea of the generic method used to identify the scheduling problem is given. In Section 3, an example is presented and the previous method is applied to precisely identify the scheduling problem. In Section 4, an integer programming model is proposed to determine the