Int. J. Production Economics 146 (2013) 199–207

Contents lists available at ScienceDirect

Int. J. Production Economics

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

A study of the robustness of the group scheduling method using an emulation of a complex FMS

Olivier Cardin*, Nasser Mebarki, Guillaume Pinot

LUNAM Université, IUT de Nantes – Université de Nantes, IRCCyN UMR CNRS 6597 (Institut de Recherche en Communications et Cybernétique de Nantes), 2 avenue du Pr Jean Rouxel, 44475 Carquefou, France

A R T I C L E I N F O

Article history:
Received 28 September 2011
Accepted 21 June 2013
Available online 11 July 2013

Keywords:
Scheduling
Robustness
Flexibility
Group scheduling
Proactive–reactive scheduling

A B S T R A C T

In the field of predictive–reactive scheduling methods, group sequencing is reputed to be robust (in terms of uncertainties absorption) due to the flexibility it adds with regard to the sequence of operations. However, this assumption has been established on experiments made on simple theoretical examples. The aim of this paper is to carry out experimentation on a complex flexible manufacturing system in order to determine whether or not the flexibility of the group scheduling method can in fact absorb uncertainties. In the study, transportation times of parts between machines are considered as uncertain. Simulation studies have been designed in order to evaluate the relationship between flexibility and the ability to absorb uncertainties. Comparisons are made between schedules generated using the group sequencing method with different flexibility levels and a schedule with no flexibility. This last schedule takes into account uncertainties whereas schedules generated using the group sequencing method do not. As it is the best possible schedule, it provides a lower bound and enables to calculate the degradation of performance of calculated schedules. The results show that group sequencing perform very well, enabling the quality of the schedule to be improved, especially when the level of uncertainty of the problem increases. The results also show that flexibility is the key factor for robustness. The rise in the level of flexibility increases the robustness of the schedule towards the uncertainties.

1. Introduction

The most classical scheduling situation is the static scheduling, also called predictive scheduling. In this context the variables, the constraints, the environment, all the data are considered as known. Then, the scheduling problem can be solved using combinatorial optimization methods, either exact methods such as linear programming, branch and bound methods, or heuristic methods (Pinedo, 2008). This phase is done off-line.

The problem is that manufacturing systems are not so deterministic. They present a lot of uncertainties and randomness, e.g., the breakdown of a machine, late material, new orders to proceed immediately, etc. During the execution of the schedule, it is frequently necessary to repair the schedule while preserving the solution's quality. This phase is done on-line and is called reactive scheduling or predictive–reactive scheduling since the reactive phase is based on a schedule computed during the predictive phase. When the schedule in progress can no longer be repaired, it is necessary to set a new predictive schedule.

Moreover, the model used to generate a schedule during the predictive phase can be more or less precise in regards with the real system (e.g. operating times considered as random, transfer times considered as negligible, etc.). Roy (2005) talks about "just about or ignorance areas that affect the modeling of a problem".

To cope with these drawbacks, one can use real-time control methods (Chan et al., 2002). These kind of methods do not make any plan and builds the schedule dynamically. Generally, these methods use priority rules which determine for each resource of the shop the next operation to proceed among the waiting operations in the resource's queue. These methods consider the real-state of the shop, so hazardous phenomena can be effectively taken into account when they occur. As these methods deal with the real-state of the system, they are called reactive methods. The major drawback of reactive methods is that their performance is generally poor.

Very few methods try to combine the advantages of both predictive and reactive methods. The proposed methods rely essentially on the introduction of slacks during the predictive phase in order to facilitate re-scheduling during the reactive phase. The idea is to introduce flexibility during the predictive phase in order to obtain a robust schedule, enabling to absorb the perturbations during its execution. This is called proactive scheduling (Herroelen and Leus, 2004; Van de Vonder et al., 2007). One

* Corresponding author. Tel.: +33 228092000; fax: +33 228092021.
E-mail addresses: olivier.cardin@irccyn.ec-nantes.fr, olivier.cardin@univ-nantes.fr (O. Cardin).

0925-5273/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ijpe.2013.06.023

of the most known proactive methods in project scheduling is the critical chain method used in several project management pieces of software (Goldratt, 1997). The method relies on the introduction of slacks on the operations of the critical path in order to absorb the delays. Another proactive method is the Just In Case Scheduling method (Swanson et al., 1994). This method identifies the probable critical delays and for each of them an alternative scheduling is proposed. Other methods, using fuzzy logic, try to capture the uncertainties in the scheduling data. Caprihan et al. (2006) have added fuzzy criteria on priority rules such as Work In Next Queue (WINQ) and Number In Next Queue (NINQ). Domingos and Polianto (2003) have proposed a procedure for on-line scheduling where decisions are based on fuzzy logic.

In this paper, we focus on the group scheduling method which is one of the most studied proactive–reactive methods. This method is very interesting because it favors the cooperation between human and machine to address the uncertainties of both the system (i.e., randomness) and the model (i.e., lack of precision). This is done by defining a partial order of the operations on each machine, permitting the decision maker to complete the scheduling decisions according to his preferences and the real state of the shop. For this, the group scheduling method builds scheduling solutions that integrate sequential flexibility; moreover the method guarantees a minimal quality corresponding to the worst case. Thus, this method brings flexibility and enables to choose in real-time the operation (among a group of permutable operations) that fits best to the real state of the system. However, group scheduling has a major drawback: when order groups are very large, the flexibility added to the schedule is bigger but the cycle times will be longer.

Group scheduling was first introduced in Erschler and Roubellat (1989). This method has been widely studied in the last 20 years, in particular in Erschler and Roubellat (1989), Billaut and Roubellat (1996), Wu et al. (1999) and Artigues et al. (2005). The flexibility added to the schedule should enable the method to be robust, i.e. the method should absorb uncertainties, randomness and lack of precision of the model. Only two precedent studies have tried to verify this supposed property of the group scheduling method. Wu et al. (1999) study the impact of disturbed processing times on the objective of weighted sum of tardiness in comparison with static and dynamic heuristics. Esswein (2003) studies the impact of disturbed processing times, due dates and release dates on a one machine problem and compares its results with a static heuristic method.

Although these two studies show the capacity of group scheduling to be robust, they are made on simple and theoretical models with simplified assumptions.

This paper presents a new study to evaluate the capacity of group scheduling method to absorb uncertainties. Contrarily to previous studies done with the group scheduling method, this study has been conducted on an emulation of a real flexible manufacturing system. As stated by Liu and MacCarthy (1996) a Flexible Manufacturing System (FMS) is an integrated system composed of automated workstations such as computer numerically controlled (CNC) machines with tool changing capability, a material handling and storage system such as automated guided vehicles or conveyors, and a computer control system which controls the operations of the whole system. It is frequently observed in the scheduling literature that transfer times are purely and simply ignored (Pinot, 2008). To evaluate the robustness of group scheduling in regards with uncertainties, we conducted experiments on a real FMS to test the effect of the non-consideration of transfer times in a predictive schedule. In order to have results on a large time horizon, experiments are made on an emulation of the real system and not directly on the real system.

The rest of the paper is organized as follows: first, the group scheduling method is presented, second, the capacity of the group scheduling method to handle uncertainties are discussed, third, the different experiments made are presented, and finally the results, very promising, are discussed.

2. Group scheduling

Van de Vonder et al. (2007) propose a classification of scheduling methods under uncertainties. Their classification relies on the way uncertainties are taken into account. From this, they identify three major categories: proactive methods, reactive methods and proactive–reactive methods.

A proactive method is a method where flexibility is added in a predictive model in order to deal with the uncertainties when they occur. Its main advantage is that it permits the computation of an optimal solution of the schedule. The problem of this kind of methods is that it deals with a prediction of the system and not with the real state of the shop.

With reactive methods, decisions are taken considering the real events of the shop. So, these methods are really taking into account the uncertain nature of manufacturing systems. They are usually called dynamic methods because they build dynamically the schedule as one goes along the events in the system. Their main advantage is that they deal with the real-state of the shop but they may give poor solutions for the schedule.

Proactive–reactive methods deal with a static schedule computed by a predictive method during a first phase of the schedule called the predictive phase. Then, in a second phase re-schedulings are done in order to take into account the real-state of the shop. This second phase is called the reactive phase where the schedule is adjusted to the uncertainties which occur in the system. Thus, these kinds of approaches are very interesting because they try to use the benefits of both the predictive and reactive methods. Bel and Cavaillé (2001) describe different proactive–reactive methods.

One of the most powerful and studied proactive–reactive methods is the group scheduling method. It is also one of the only proactive–reactive methods to be implemented in an industrial scheduling software named ORDOSoftware (Roubellat et al., 1995).

The group scheduling method was first introduced in Erschler and Roubellat (1989). The goal of this method is to provide a sequential flexibility during the execution of the schedule and to guarantee a minimal quality corresponding to the worst case. This method has been developed in the last 20 years, in particular in Erschler and Roubellat (1989), Billaut and Roubellat (1996), Wu et al. (1999) and Artigues et al. (2005). A full theoretical description of the method is available in Artigues et al. (2005). Let us describe the method and for this introduce some notations: in the rest of this article, the index i will be used for the operations and the index l will be used for the resources.

A group of permutable operations is a set of operations to be performed on a given resource M_l in an arbitrary order. It is named $g_{l,k}$. The group containing the operation O_i is denoted by $g(i)$. The index k will be used for the sequence number of a group.

A group sequence is defined as a sequence of ν_l groups (of permutable operations) on each machine M_l: $g_{l,1},\dots,g_{l,\nu l}$, performed in this particular order. On a given machine, the group after (resp. before) $g(i)$ is denoted by $g^{+}(i)$ (resp. $g^{-}(i)$).

A group sequence is feasible if for each group, all the permutations among all the operations of the same group give a feasible schedule (i.e. a schedule which satisfies all the constraints of the problem). As a matter of fact, a group sequence describes a set of valid schedules, without enumerating them. The quality of a group sequence is expressed in the same way as that of a classical