ELSEVIER

European Journal of Operational Research 139 (2002) 230–244

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/dsw

Production, Manufacturing and Logistics

Using real time information for effective dynamic scheduling

Peter Cowling [a,*], Marcus Johansson [b]

[a] *Department of Computing, University of Bradford, Bradford, West Yorkshire, BD7 1DB, UK*
[b] *Division of Manufacturing Engineering and Operations Management, University of Nottingham, Nottingham NG7 2RD, UK*

Received 28 February 1999; accepted 02 May 2001

Abstract

In many production processes real time information may be obtained from process control computers and other monitoring systems, but most existing scheduling models are unable to use this information to effectively influence scheduling decisions in real time. In this paper we develop a general framework for using real time information to improve scheduling decisions, which allows us to trade off the quality of the revised schedule against the production disturbance which results from changing the planned schedule. We illustrate how our framework can be used to select a strategy for using real time information for a single machine scheduling model and discuss how it may be used to incorporate real time information into scheduling the complex production processes of steel continuous caster planning. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Scheduling; Production; Rescheduling; Real time information

1. Introduction

Although scheduling is a well researched area, and numerous articles and books have been published, classical scheduling theory has been little used in real production environments (Stoop and Wiers, 1996). We believe that scheduling research has much to offer industry and commerce, but that more work is needed to address the "gap" between scheduling theory and practice (MacCarthy and Liu, 1993). One frequent assumption of scheduling theory, which rarely holds in practice, is that the scheduling environment is static. In many production and service systems, schedules must be revised frequently in response to both instantaneous events, which occur without warning, and anticipated events where information is given in advance by, for example, process control computers or customers. In this paper we develop a framework for handling real time information concerning anticipated future events. Note that in this case the time of arrival of the information is important in deciding the best schedule revision strategy to adopt.

Many manufacturing environments use an material requirements planning (MRP), manufacturing resources planning (MRPII) or enterprise resources planning (ERP) system for medium term planning. Such a system divides the planning

* Corresponding author.
E-mail addresses: peter.cowling@scm.brad.ac.uk (P. Cowling), marcus.b.johansson@uk.andersen.com (M. Johansson).

horizon into discrete time buckets and requires a medium term production plan for several future time buckets, which is used to provide due dates and release dates for detailed production scheduling. We gain the advantage of being able to divide a complex medium term capacity planning problem into smaller and more manageable pieces at the cost of high rigidity in the production plan. The question as to how a scheduler should respond to changes in a dynamic system in this environment is a fruitful area for research. Between schedule creation and execution one or several assumptions may have changed concerning, for example, machine availability or material supply. The obvious question is how to react to this type of information and how different scheduling strategies affect the original plan or sequence of jobs? To answer this question one needs to consider effects on both upstream and downstream operations as well as the effects on longer term plans. This has previously been difficult, since there has been a lack of real time information concerning system status. However, in many current production and service systems a great deal of real time information is captured for control and monitoring purposes. In deciding how to react we must consider not only the quality of the revised schedule but also the disruption caused by schedule revision. Our framework will consider the trade off between these two factors.

Scheduling research has failed to keep pace with technological developments in process control and monitoring systems. In this paper we present a framework for effectively incorporating real time information produced by process control and monitoring systems into scheduling models and in so doing to address an important aspect of the "gap" between scheduling theory and practice.

In Section 2 we consider the types of real time information encountered in practice and survey the literature on dynamic scheduling. Here we present our notions of schedule repair and rescheduling. In Section 3 we present two measures for determining the value of real time information, *utility* which measures the improvement in our original scheduling objectives due to schedule revision and *stability* which measures the disruption caused by schedule revision. In Sections 4 and 5 we apply our notions of utility and stability to the simple $n/1//\overline{C}$ single machine scheduling model. Section 4 investigates in detail the response of the model to a single piece of real time information. In Section 5 we carry out a simulation study to investigate the behaviour of several strategies for dealing with multiple pieces of real time information. In Section 6 we discuss how our ideas may be applied to the complex scheduling environment of the steel continuous caster, where a great deal of information is produced by process control computers. Conclusions are presented in Section 7.

2. Using real time data

Historically, one of the major reasons for uncertainty in real scheduling environments has been the lack of accurate information (Ovacik and Uzsoy, 1994, 1997). However, the advent of computerised information systems capable of tracking job and machine status in real time has changed this situation. In many of the process industries, including the steel making example, which we will discuss later, information is generated in real time by process control computers. In discrete parts manufacture, computer systems for the entry and dissemination of data, such as VDU terminals and bar code scanners, are placed at various locations on the shop floor, to record information concerning the location and status of jobs and resources and to display this information for control purposes. Feedback can be generated from several or all work centres to track jobs and update their progress. This technology is now comparatively cheap and very effective (Singh, 1996).

Real time information is commonly used to improve estimated values of some parameters, such as processing time or worker performance, based on larger sample sizes. Real time information is only rarely used to improve schedules and then only in an ad hoc manner to locally correct short-term problems which might arise due to machine failure, etc. In this paper we develop a systematic approach to the use of real time information in scheduling.

Lindau et al. (1994) and Fredendall et al. (1996) have made empirical studies on the impact of real time information for specific industrial scheduling models. Both studies show that the performance of a system without shop floor information is inferior to a system where real time information is used to make real time scheduling decisions. Ovacik and Uzsoy (1994) exploit shop floor information to consider not only the jobs available at the machine at the time of the scheduling decision but also jobs that are going to become available within a certain time window. They studied a dynamic single-machine problem with sequence-dependent set-ups by comparing heuristic rules that use global information in making local scheduling decisions at the machine level to several myopic dispatching rules which use only local information. Chang (1997) considered a dynamic job shop where the arrival and nature of jobs is governed by known probability distributions. He showed that if estimates of queue length are updated in response to real time information, then the performance of several dispatching rules could be improved.

A dynamic scheduling system is one that uses real time information as it arrives. The planning and scheduling process then consists of one or more nested feedback loops, where each feedback loop corresponds to a scheduling period (month, week, day) and some group of processes which are undergone by the jobs. This group of processes might involve something as simple as being processed on a single machine, right up to a full manufacturing process. First, we formulate a static schedule for each period. Then we obtain real time information concerning, for example, the progress of each job and the shop floor situation, and react to that information to revise the schedule for the current period and processes, when circumstances make schedule changes desirable or necessary. Each feedback loop defines a *dynamic scheduling* problem. Each of our dynamic schedules will interact with other dynamic schedules at different time horizons and upstream and downstream processes, so that the effects of modifying a local schedule in response to a given piece of real time information must be considered throughout the system.

Practical scheduling systems need to be able to react to significant real time events within an acceptable response time and revise schedules appropriately. *Rescheduling* occurs when we restart the scheduling process from scratch. *Schedule repair* refers to some local adjustment of the current schedule and may be preferable for many reasons, not least because feasible schedules may be difficult to generate within acceptable time limits in many environments. The practical importance of the decision whether to reschedule or repair has been noted in recent papers by Lee et al. (1996) and Dorn and Kerr (1994). In order to decide what action we should take in response to an event, we should have some idea of the value of modifying our current schedules in response to the event and some measure of the overall impact of making schedule changes. In the following section we will use the quantitative measures of *utility* and *stability* to assess the value and impact of schedule changes.

Schedule repair plays an important role in some knowledge-based systems which have been developed in the Artificial Intelligence community. ISIS (Fox and Smith, 1984), OPIS (Smith et al., 1990) and IOSS (Park et al., 1996) are systems which have used knowledge-based scheduling methods to generate a feasible schedule and interactive scheduling methods to revise the existing schedule. CABINS (Miyashita, 1995) is an intelligent scheduling system which integrates case-based reasoning mechanisms for incremental accumulation and re-use of past schedule repair experiences to achieve efficiency of the revision process while preserving the quality of the resulting schedule. Suresh and Chaudhari (1993) survey several other knowledge based systems in this area.

Some work on schedule repair based on heuristic rules shows that schedule repair has the potential to improve the efficiency and flexibility of scheduling systems. Zweben et al. (1994) use simulated annealing and constraint propagation to repair schedules for space shuttle ground operations. Efstathiou (1996) introduced a software package, developed at Rover, to help schedulers carry out manual or semi-automatic schedule repair in response to real time events.

Work on dynamic scheduling is surveyed in the paper of Suresh and Chaudhari (1993) and, more recently, in the thesis of Guo (1999). When machine failure requires rescheduling within a job shop or flow shop environment we may use the match-up scheduling approach, which has been considered in several papers including Akturk and Gorgolu (1999) and Bean et al. (1991). When machine failure requires revision of the current schedule, this revision is carried out subject to ensuring that the revised schedule "matches up" to the original schedule as soon as possible after the machine breakdown, allowing some consideration of the stability of the shop. Jain and Elmaraghy (1997) used genetic algorithms to obtain an initial schedule and then heuristic rules to handle shop floor disruption. Daniels and Kouvelis (1995) and Leon et al. (1994) discussed the concept of schedule "robustness". Here we consider how adverse the effects of our chosen schedule repair strategy may be in response to machine failure in order to design a robust initial schedule to minimise these effects. None of the above work deals with the trade off between schedule quality and schedule stability in choosing an appropriate schedule repair strategy. Wu et al. (1993) consider this trade off for events taking place in real time, in order to compare the performance of three schedule repair strategies. All these approaches consider the best way of dealing with events as they occur, rather than the arrival of real time information concerning anticipated future events, where the time of arrival of the information is critical to the way in which the information may be effectively handled.

Ehlers and Van Rensburg (1994) consider eight different types of real time information. Such a taxonomy may be useful but we consider that there are essentially two kinds of real time information, illustrated in Fig. 1: that relating to the status of resources and that relating to the status of jobs.

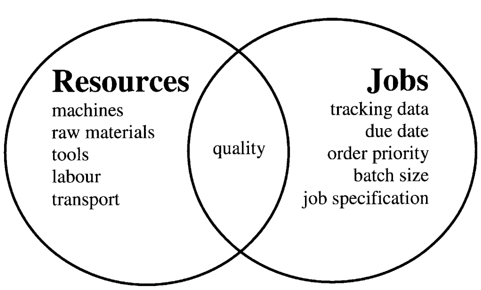

Fig. 1. Classification of real time information.

Real time information relating to the status of resources includes information concerning machines, raw materials, tools, labour, etc. Real time information relating to the status of jobs includes tracking data for each operation, data concerning successfully completed processing stages and information about schedule adherence. Information on actual or potential disruptions may relate to resources or jobs. Machine breakdowns, material or tool shortages and longer-than-expected processing times give resource problems. Job related disturbances arising from planning systems and customers include changes in priority, reassignments of jobs to orders and the emergence of new jobs. Quality problems may relate to both resources and jobs.

By having well-defined procedures for handling real time data we may both reduce the nervousness of the system and opportunistically improve schedule performance, compared with using ad hoc approaches. Most particularly, we can make a priori decisions as to what levels of system disruption are tolerable for a given level of performance improvement. When real time data arrives it is put through a four stage process: *detection*, *classification*, *identification* and *diagnosis*. Since real time events may occur every few minutes in a system (Stoop and Wiers, 1996) it is important that the procedure is standardised, with automatic computer intervention where possible. *Detection*: real time data are detected by, for example, sensors, barcode scanners or operators. Understanding the detection process will lead to effective use of real time data capture devices, and removal of unnecessary and useless devices. *Classification*: we must classify the event and decide whether it may be handled automatically, or requires a human decision-maker. *Identification*: after the real time information has been classified and possibly dealt with automatically, there will often remain a need for a more detailed analysis of the disturbance type. For example, we may wish more information as to why a machine breakdown has occurred, e.g. lack of maintenance or sabotage. Frequently occurring types of disturbance need deeper investigation, both for prevention and improved prediction. *Diagnosis*: here we decide what action to take in response to the piece of real time information. It is possible that we should take no action, conduct a limited repair or reschedule from scratch. We will use the quantitative measures of utility and stability to help us make this decision as to what action should be taken.

3. Measures of utility and stability

If we have a range of good techniques for repairing schedules and rescheduling in the presence of real time information, we must still address the important issue of whether to repair or reschedule and which schedule repair or rescheduling strategy should be used in response to any given collection of real time information. When we receive information concerning a highly significant anticipated future event, such as unplanned machine maintenance, the most appropriate course of action may well be to discard the current schedule and reschedule from scratch. In doing so, however, we must take into account the impact that this decision will have on current schedules for upstream and downstream processes and on future plans and schedules. When information concerning less significant anticipated future events is received, it may be most appropriate to adopt a schedule repair strategy, which attempts to find a revised schedule while minimising the disturbance to current and future plans. This "wait and see" approach means that we may do nothing or carry out only a small local repair in response to events. Of course we find, at some stage, that the accumulation of small anticipated events mean that the disruption caused by rescheduling is justified. It is important to note that in any operation where there are schedules which may be subject to unforeseen change, there must necessarily be a strategy for dealing with these events. Our experience of the steel, furniture and paper industries suggests that the strategies which are used in industry are, however, often ad hoc and not subject to the same kind of analytical rigour or sophisticated techniques which are applied to the scheduling decisions themselves.

The practical importance of the decision whether to do as little as possible, locally repair the schedule or to reschedule from scratch in response to a real time event is identified in Lee et al. (1996) and Dorn and Kerr (1994). Both these papers discuss decision support for scheduling of primary steel making processes, where a great deal of real time process control information is available.

In this section we will define two measures to guide the decision as to what strategy should be used to repair a schedule in response to real time information, *utility* and *stability*.

Utility will measure the benefit which may be gained by using a particular rescheduling strategy. Suppose that we have a mathematical model M of a scheduling process, where we have n numerically defined objective criteria $(O_1, O_2, \ldots, O_n)$. Without loss of generality we suppose that each objective criterion is to be maximised. For example a piece of real time information arising from process control computer might be that upstream process controllers report that "the true size of job A123 is 50% greater than the scheduled size". Clearly the value of this information for rescheduling purposes depends heavily on the time at which it arrives. We suppose that for a (potentially compound) piece of real time information E there is a strategy S_0, corresponding to the notion of "do nothing", which yields objective function values $(a_1, a_2, \ldots, a_n)$. For example S_0 might correspond to the strategy "make the originally scheduled orders to stock in response to the customer order cancellation in the hope of a later sale" or "create a buffer of work to be done in front of the malfunctioning machine and leave other machine schedules unchanged", with this latter strategy corresponding to the *pushback* strategy of Bean et al. (1991). A further example is given in the following section. Suppose further that we have a strategy S_1 which will produce a unique solution for the scheduling problem modified by this real time information, yielding objective function values $(b_1, b_2, \ldots, b_n)$. Then the utility U is the multiple valued function given by

$$U(M, S_0, S_1, E) = (b_1 - a_1, b_2 - a_2, \ldots, b_n - a_n).$$

When the model M has only a single objective criterion we will have a strategy S_{opt} which will give an optimal solution in response to the real

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.