Available online at www.sciencedirect.com

SCIENCE DIRECT®

European Journal of Operational Research 161 (2005) 86–110

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/dsw

Executing production schedules in the face of uncertainties: A review and some future directions

Haldun Aytug [a,*], Mark A. Lawley [b], Kenneth McKay [c], Shantha Mohan [d], Reha Uzsoy [e]

[a] Department of Decision and Information Sciences, Warrington College of Business Administration, University of Florida, Gainesville, FL 32611, USA

[b] School of Industrial Engineering, 1287 Grissom Hall, Purdue University, West Lafayette, IN 47907-1287, USA

[c] Faculty of Business Administration, Memorial University of Newfoundland, St. John's, NF, Canada A1B 3X5

[d] Kaveri Inc., Software Technology and Management Consulting, 261 Parkside Drive, Palo Alto, CA 94306, USA

[e] Laboratory for Extended Enterprises at Purdue, 1287 Grissom Hall, Purdue University, West Lafayette, IN 47907-1287, USA

Available online 21 November 2003

Abstract

We review the literature on executing production schedules in the presence of unforeseen disruptions on the shop floor. We discuss a number of issues related to problem formulation, and discuss the functions of the production schedule in the organization and provide a taxonomy of the different types of uncertainty faced by scheduling algorithms. We then review previous research relative to these issues, and suggest a number of directions for future work in this area.

© 2003 Elsevier B.V. All rights reserved.

Keywords: Production scheduling; Rescheduling; Structural control

1. Introduction

Manufacturing operations can be faced with a wide range of uncertainties and production control is charged with accommodating these in advance or reacting after the fact. There may be relatively little uncertainty, or a plant may experience pervasive and rampant chaos. When there are large amounts of uncertainty, Emerson's description may still be appropriate:

> ... but most of the industrial plants of the world are still in the stage of civilization of which as to transportation the old freight wagons and prairie schooners across the plains were types. They started when they got ready, they arrived some time, and nobody knew where they were nor what route they were taking in between. (Emerson, 1913; p. 251).

* Corresponding author. Tel.: +1-352-3922468; fax: +1-352-3925438.

E-mail addresses: aytugh@ufl.edu (H. Aytug), malawley@ecn.purdue.edu (M.A. Lawley), kenmckay@mun.ca (K. McKay), shantha_mohan@alumni-mail.gsia.cmu.edu (S. Mohan), uzsoy@ecn.purdue.edu (R. Uzsoy).

0377-2217/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2003.08.027

Uncertainty and the disruptions associated with the resulting perturbations have been topics of discussion since the early 1900s. For example, Gantt (1919) is known for what we call the Gantt Chart today, but he developed several different charts and the one that he considered the most useful was not the planning chart, but the chart prepared by the floor workers (operators or supervisors) providing feedback to the planners and schedulers—why the plan and schedule did not execute as planned. They reported back causes of delays, yield problems, and so forth associated with material, tools, and machinery. An early description of the scheduling task explicitly noted that the planners had to anticipate future difficulties and discount them (Coburn, 1918). Disruptions and uncertainty have been a problem since the beginning of systemized manufacturing and remain so today.

There has been an extensive body of research on production scheduling problems since the original mathematical formulation of these problems in the late 1950s. These formulations typically involve the assignment of scarce resources, usually machines, to competing tasks over time to optimize some aspect of system performance either exactly or approximately. This literature can be broadly classified into two main areas: deterministic scheduling research, where all problem parameters are assumed to be known with certainty, and stochastic scheduling, where at least some parameters are random variables. Much of the stochastic scheduling work has assumed that all parameters are random variables, and has thus focused on local control policies such as dispatching rules aimed at minimizing some measure of performance in the expectation. Most of these methods seldom use any information about the global state of the shop, or try to create a schedule for the entire shop prior to its execution. In deterministic scheduling research a larger view is taken and multiple machines are often modelled. The deterministic approach is to plan the work through the machines over a period of time in the best way possible given a specific objective to optimize. The implicit assumption here is often that a schedule can be executed directly as developed. However, in recent years many authors have recognized that this is an unlikely scenario in many manufacturing environments, and have made efforts to extend the deterministic approaches to situations with some form of uncertainty. The basic assumption in much of this work, which forms the focus of this paper, is that a system that works in a deterministic environment can be engineered to work under at least certain stochastic conditions.

A pervasive assumption in the deterministic scheduling field has been that the schedule once released to the production floor can be executed as planned. However, many production systems are subject to executional uncertainties that prevent the execution of a production schedule exactly as it is developed. Examples of such disruptions include machine failures, quality problems, arrival of urgent jobs and a myriad of other possibilities. Theoretical scheduling research also typically fails to consider the organizational discipline needed to execute a schedule correctly. Thus, for example, the specific incentives used for the shop-floor personnel may cause them to override the schedule, in effect introducing another type of uncertainty. The inability of much scheduling research to address the general issue of uncertainty is often cited as a major reason for the lack of influence of scheduling research on industrial practice. Although in recent years there has been a steadily increasing volume of research in this area, we believe there are several different approaches that have developed largely in isolation, and need to be evaluated and discussed together to provide a broad perspective on this important problem area.

For the purposes of this paper, we shall restrict ourselves to the type of scheduling problem encountered in manufacturing environments, where the basic problem is to allocate machines, and perhaps other resources such as tooling or operators, to jobs in order to exactly or approximately optimize system performance. Hence we shall ignore a number of other decisions, such as order release, due date setting and lot sizing, which are often considered part of the larger production planning decision, and whose solutions clearly affect the scheduling function. We shall use the term "schedule" to denote an assignment of machines to jobs for a specific period into the future, referred to as the schedule horizon. We shall also be primarily concerned with the issue of how to execute a given schedule on the shop floor in the presence of uncertainties, rather than discussing how different types of schedules can be constructed in the first place. Finally, we shall focus our review primarily on work over the last decade, since much of the earlier work has been reviewed by other authors (e.g., Harmonosky and Robohn, 1991; Suresh and Chaudhuri, 1993; Szelke and Kerr, 1994). However, in contrast to the previous review papers, which tend to use a taxonomy of research based on solution techniques (conventional, artificial intelligence, etc.), we try to develop a taxonomy based on the formulation of the scheduling problem used. We make no pretence to have provided an exhaustive list of references in this rapidly expanding research area, but we feel we have provided a synthesis of the main research directions.

We believe that in order to understand the different issues involved in developing effective scheduling methods for environments with executional uncertainties, one needs to examine the ways in which the organization uses the production schedule—in other words, why a schedule is necessary or helpful in the first place. We then introduce and discuss a taxonomy for viewing and classifying production uncertainties. Following these discussions, we can examine a number of the different problem formulations in the scheduling literature and discuss their various strengths and weaknesses. We then examine issues associated with schedule execution in automated settings when uncertainty exists. We conclude with suggestions for future research.

2. Purposes of scheduling

There are a number of reasons why a manufacturing organization might want to develop a production schedule for some time period into the future. Younger, in perhaps the first book dedicated to scheduling, posed it this way:

> Well-organized and carefully executed work routing, scheduling, and dispatching are necessary to bring production through in the required quantity, of the required quality, at the required time, and at the most reasonable cost. (Younger, 1930, p. iii).

These goals are the highest level and provide the most obvious reasons why scheduling is performed. More specifically, Reinfeld who was instrumental in the founding of the American Production and Inventory Control Society (APICS), viewed the problem as:

> Production control is the task of predicting, planning and scheduling work, taking into account manpower, materials availability and other capacity restrictions, and cost so as to achieve proper quality and quantity at the time it is needed and then following up the schedule to see that the plan is carried out, using whatever systems have proven satisfactory for the purpose. (Reinfeld, 1959, p. 66).

In the simplest of terms, these statements provide a basis for the mathematical formulations and computer systems that create schedules. They both note the cost objectives, quality goals, delivery concerns, and quantity targets. These are the technical or mechanical aspects of scheduling. The basic assumption when one develops a production schedule is that this will serve as an instruction to the shop floor, causing the shop to execute events in the sequence and timing suggested in the schedule. In many systems, this involves developing a schedule under certain assumptions as to the execution environment (most commonly, that no disruptions will occur) and releasing it to shop-floor personnel to guide their decisions. We will refer to this type of schedule as a *predictive schedule*. In an environment with tightly integrated automation, it may well be that the predictive schedule drives execution directly by interacting with machine controllers and other system components. Wiers (1997) provides a taxonomy and categorization schema for matching solution methods (e.g., decision support systems, optimization code, etc.) to a situation based on the degree of uncertainty—the tightly integrated automated situation being called a *smooth* shop while a dynamic open job shop with significant uncertainty is called a *stress* shop.

In almost any environment other than what Wiers would categorize as a smooth shop, it is highly unlikely that the predictive schedule will be executed exactly. Disruptions will require modifications to permit execution, and perhaps create opportunities to improve shop performance based on the situation encountered after a disruption. Hence, there may be substantial deviations from the predictive schedule over the course of its execution due to unforeseen disruptions such as machine breakdowns or shop-floor personnel overriding the predictive schedule. The process of modifying the predictive schedule in the face of executional disruptions is generally referred to as *reactive scheduling* or *rescheduling*. The nature of the schedules developed in reaction to disruptions depends on the nature of the realized disruptions and the capabilities of the execution agent reacting to them. The reaction generally takes the form of either modifying the existing predictive schedule, or generating a completely new schedule that is followed until the next disruption occurs.

The following subsections explore other purposes of scheduling related to uncertainty.

2.1. As a capacity check for higher-level reasoning

In this situation a higher-level production planning system will verify that it has the capacity to produce the planned work over a given time period by developing a complete, finite-capacity production schedule and making allocations of resources to jobs at specific points in time. Dauzere-Peres and Lasserre (1994) give an example of this approach. In industrial practice, there are several systems that perform the capacity check by performing a deterministic simulation of the production system in question (Pritsker and Snyder, 1997). Sun and Lin (1994) present an interesting approach in this line based on considering backward scheduling from due dates at the level of individual operations. The backward scheduling problems are solved using a rolling horizon procedure, but there is limited experimental testing of the effectiveness of the approach. A number of the advanced planning and scheduling (APS) systems on the market today also use this approach (Musselman and Uzsoy, 2001). Note that in this case, there is not necessarily a thought that the schedule will ever be executed as developed—the purpose is to verify that there exists at least one capacity-feasible resource allocation for the work planned over this period.

In practice, planners do a capacity check to also identify peak load intervals and lower periods. The peak load situations become critical when uncertainty increases as there are fewer degrees of freedom for recovery. Any elective actions (i.e., planned activities) that may increase uncertainty are routinely moved away from these peak capacity zones. The lower load zones have a greater potential for absorbing uncertainty and the planners schedule high-risk work for these periods. The peak zones are also identified as being of interest if critical work is planned—backup plans are prepared and some activities put into action just-in-case. Not all planners and schedulers do this type of reasoning, but some do (McKay et al., 1995).

2.2. To provide visibility of future plans within the shop

A major function of the production schedule, which we feel is often overlooked in the research community, is that of providing visibility of future actions for the rest of the organization, and for internal and external suppliers and customers. The production schedule may serve to identify potential capacity conflicts at critical resources, permitting management to take action to avoid them. In many ways, it allows the astute shop-floor manager to organize production resources to best support smooth schedule execution. We often hear from industrial practitioners that production systems gain a certain momentum, and that violent schedule changes throw the floor into confusion. We believe that this refers to the effects of using the schedule for visibility. Shop-floor personnel will use the schedule to guide their actions, positioning work, tooling and operators in a way that will smooth the execution of the schedule. They will look at the schedule for situations implying resource conflicts or tight constraints and orchestrate the situation before, during, and after the expected event (McKay et al., 1995). A significant schedule change may thus require a significant amount of reorganization on the shop floor as machines are retooled, operators reassigned and work in progress interrupted to bring the system in line with the new schedule. We shall refer to this process of bringing resources in line with a new schedule as *system reconfiguration*. The greater the number of tightly coupled decisions, the more difficult the reconfiguration. By extension, this logic applies to customers of the shop in question who plan their activities based on the planned completion times of activities in the original schedule that has now been changed.

This objective of providing visibility has gained new importance with the trend towards increasing collaboration between the elements of supply chains. Enabled by Internet technology, it is rapidly becoming common for companies to share their production schedules with their suppliers on a continuous basis, with the expectation that the suppliers will use this information to provide services such as just in time material delivery. In this environment, changes to the production schedule at a downstream node of the supply chain can cause significant disruptions of upstream operations. The potential impact of such disruptions can be quite high, as evidenced by the well-studied "bullwhip effect" (Chen et al., 2000) in supply chains, that causes variation at downstream nodes in the supply chain to be amplified at upstream stages.

2.3. To provide degrees of freedom for reactive scheduling

When there is a period of uncertainty and instability, it is important to have capacity on the resources that have the greatest capability for resolution and re-stabilization (McKay et al., 1995). Schedulers and planners first try to lock in or assign resources that have low flexibility and for which there are few alternatives. This allows the scheduler to use the more flexible resources to solve the hard problems as schedules get tight. This also provides the planner the proverbial Swiss Army Knife when a problem occurs and some spare capacity exists on the flexible resource. The objective of this type of scheduling is to react without affecting large portions of the factory and causing chain-reactions. If the key resource is committed early on, pre-empting that resource to help with reactive problem-solving can cause a ripple effect throughout the schedule and plant organization.

2.4. To evaluate performance

Another potential use for a schedule is to provide a yardstick by which to measure the performance of shop-floor personnel. In this situation, a predictive schedule is used to set goals which the shop-floor personnel should achieve. The performance of shop-floor personnel is evaluated at the end of one or more planning periods using the deviations of the historical schedule from the predictive. This use of schedules is important in that it affects the way shop-floor personnel will react to unforeseen disruptions, influencing the evolution of the historical schedule as distinct from the predictive. Najmi and Lozinski (1989) give an example of this use of predictive schedules. When using a predictive schedule for this purpose, it is important that all parties sign off on the plan as feasible and doable when issued. It is not unusual for a predictive plan to be totally unrealistic and political in nature—neither feasible nor reasonable. Such a questionable plan should not be used to measure performance.

Shop-floor personnel are not the only targets for evaluation. Gantt (1919) discusses how a schedule or plan can be used to gauge the performance of management (as distinct from the floor personnel). Management's job was to create a situation in which the worker could do the desired work at the time desired and with the desired results. If tooling was not ready—it was a management issue. If material had not arrived—it was management's fault. The job of management was to coordinate and manage the resources so that execution was possible. Although scheduling and scheduling feedback could be used for management evaluation, we have not encountered this specific use in any of our field work.

2.5. To avoid further problems

In a strictly reactive situation without any feed forward control, the future is taken as it occurs—everything is a surprise and there is no active mitigation of possible side-effects and problems. McKay et al. (1995) showed that a major portion of the scheduler's sequencing and scheduling tasks related to minimizing the impact of future events whose occurrence was considered highly probable. Some events might be in the immediate future (e.g., did the technician successfully repair the wave soldering machine?) or elsewhere on the planning horizon (e.g., a machine upgrade is scheduled for next month, will it be successful). This type of special sequencing can be considered a conservative or risk-averse posture in which a short term sub-optimization strategy is used to achieve a greater system level performance. The underlying assumption is that if the impact occurs as predicted, the amount of work to resequence, the amount of rework, the amount of lost value-added activities will be reduced. Without the special decision process analysing the uncertainty, the full force of the perturbation occurs.

The number of different potential uses for a production schedule underscores the need for a satisfactory execution of the scheduling task, or at least a feasible plan for the future. To be satisfactory or feasible the schedule must address uncertainty. On the other hand, the diversity of the groups affected by the schedule also makes the task of measuring schedule quality more difficult, since the schedule is used for different purposes by different groups, which are often trying to achieve different goals. Gary et al. (2000) discuss this issue in detail. The inclusion of uncertainty and how to measure the quality of uncertainty inclusion makes the measurement of schedule quality even harder.

Given these purposes of the scheduling activity, we can now consider the various types of uncertainty that may be encountered in a scheduling environment. In the following section we provide a taxonomy of executional uncertainties that will allow us to put existing work in perspective.

3. A taxonomy for uncertainty

In Section 1, we briefly outlined five purposes of scheduling beyond that of simple resource allocation and sequencing and discussed how the purposes focus on the meaning of uncertainty to a plan and planner. To a human planner and to those interpreting a plan on the factory floor, uncertainty is not simply an independent stochastic concept confined to one parameter of the problem. Uncertainty in a real manufacturing situation is a complex phenomenon. Variability in processing speed has a different impact on the situation if the variation occurs early in the day or close to the end of a shift. Uncertainty that affects yield is more important after a few operations when value has been added and replacing the scrapped material in time to meet a due date is difficult, as opposed to yield variation in the very first operation. Uncertainty that affects material availability may be more important than time impacts—sometimes, sometimes not. Operator performance may be more uncertain just before or after a long weekend than mid-week. Uncertainty experienced on the night shift may have more impact than the same uncertainty encountered during the day when additional support staff and management are available for problem-solving. Hence, when the term uncertainty is used, what is meant? What does the uncertainty mean to the situation? How does the specific type of uncertainty affect predictive and reactive scheduling? What type of uncertainty does a specific modelling approach address? What type of uncertainty impact is incorporated? In this section, we introduce a preliminary taxonomy that can aid in organizing scheduling research including uncertainty. We suggest that the explicit consideration of uncertainty—how it arises, what it means, what the immediate and long term impacts are, and what the interdependencies are—should affect the problem formulation and solution processes we use in addressing the scheduling problem. While it is clearly impossible (and maybe even undesirable) to explicitly address all conceivable sources of uncertainty in a scheduling decision, it is essential that the most significant be considered for even reasonably successful execution to be possible.

There appear to be three key dimensions of uncertainty—cause, context, impact—that can help to categorize problem formulations and processes. For example, *cause* may be tooling not available, the *context* is the bottleneck machine on Monday morning, and the *impact* is a delay in setup—the machine cannot start when expected. Given this framework, it is possible to consider different options for reactive scheduling, and to consider how existing research addresses the reality of the situation. To explain the meaning of cause, context, and impact, the dimensions can be further decomposed.

Cause can be viewed as *object* (e.g., material, process, resource, tooling, personnel) and *state* (e.g., ready, not ready, high quality, low quality, damaged, healthy). As we will discuss in later sections, the majority of current scheduling research focuses on resource oriented uncertainty—variations in processing times, mean time between failure and mean time for repair. We are not aware of research that explicitly models and discusses such uncertainty causes as lower than expected material quality (material is typically modelled as good or bad, not degrees of goodness which is possible in the real world).

Context refers to the environmental situation at the time of the scheduled event (e.g., nothing special, resource just repaired or upgraded, when in week or day or shift (if it matters), experience or training of the crew). Essentially, is there anything about the context that would alter expectations for processing time, yield, or some other performance metric? The situation is either *context-free* or *context-sensitive*. A context-free situation would require no additional information or special decision making while a context-sensitive formulation would have information about the context and associated implication. The majority of research is context-free. That is, each day or time interval is viewed as the same as any other and so forth. Recent research such as O'Donovan et al. (1999), McKay et al. (2000), and Black (2001) include context information in the modelling and can be considered context-sensitive.

Impact refers to the result of the uncertainty. Is the impact a shortened or elongated processing time? Does the uncertainty affect the starting and finish times for setup? Does the uncertainty impact one or multiple resources? Does the perturbation impact material availability, or direct product cost? Is only one job impacted or do interdependencies exist that result in multiple tasks being affected? In general terms, the impact can be categorized as *time*, *material*, *quality*, *independent* or *dependent*, and *context-free* or *context-sensitive*. Independent and dependent refer to possible relationships to other jobs and activities. Research has typically focused on independent job streams in which there are no cross constraints and relationships between the work being scheduled. Context is also present on the impact side—not all jobs or situations react the same way to the same perturbation. Context-free is when one assumes that jobs do not vary in their response. The aversion dynamics heuristic (McKay et al., 2000; Black, 2001) is an example where the impact is time and the jobs are context-sensitive. The aversion dynamics concept uses information about a resource's recovery from an event and the sensitivity of the work to perceived risk to dynamically suboptimize and take a conservative posture around an event. These types of information are contextual and limit the generality of the heuristic, but capture an important element of the real-world scheduling problem.

3.1. Inclusion

The cause, context, and impact dimensions are on the problem side. The existence or orientation of these characteristics in existing research can be categorized accordingly. There are two other questions that can help organize the research results—Are the factors explicitly taken into account when crafting the predictive schedule? or Are the factors accommodated in some fashion when rescheduling is performed? For example, the placement of intelligent slack is an example of the former (O'Donovan et al., 1999) and the Averse-I heuristic in aversion dynamics is an example of the latter. Averse II and III in Black (2001) show

how to use aversion dynamics in a predictive fashion.

In summary, uncertainty is a large and complex topic and some form of categorization schema is warranted. We have proposed four preliminary dimensions that may be suitable for this purpose. The taxonomy is:

- Cause—object, state;
- Context—free or sensitive;
- Impact—time, material, quality, dependency, context;
- Inclusion—predictive and/or reactive.

These different aspects of uncertainty should be considered in terms of problem definition. However, any optimization approach to scheduling must also consider the costs imposed on the system by the disruptions.

There are at least two main types of costs that need to be considered—those related to the performance of the system against whatever conventional scheduling criteria, such as due date performance or flow time, are being considered, and those relating to the costs of system reconfiguration due to reacting. The costs of system instability and reconfiguration can be considered from three perspectives:

(i) costs that are incurred in anticipation of the disruption but which are wasted since the disruption does not occur,
(ii) costs incurred in anticipation of the uncertainty when the perturbation takes place,
(iii) costs incurred after or during the perturbation as the system is reconfigured.

McKay et al. (2000) and Black (2001) investigated tardiness issues when conservative postures are taken when uncertainty is perceived. They find that sub-optimization for a limited time offers significant benefits when the problem occurs and only minor penalties when the prediction is wrong. Mehta and Uzsoy (1998, 1999) have shown that taking a conservative approach to completion time estimation by using safety lead time buffers derived from statistical information on machine failures provides significantly improved completion time estimates at the cost of minimal degradation in more conventional measures related to due date performance.

We would suggest that the approach to take regarding a scheduling problem with executional uncertainties depends to a great extent on the robustness and reactive capabilities of the situation being scheduled and the degree of independence that exists in the factory. In research and in scheduling technology, dispatching based scheduling procedures that assign jobs to machines dynamically as machines and jobs become available and that provide very little visibility into the future are prevalent. This suggests that in many manufacturing environments reconfiguration costs are at least perceived to be negligible (at least by developers and researchers) and that almost everything is considered to be independent and robust. If this is the case, then the need to include uncertainty characteristics in the modelling is minimized. On the other extreme, in an environment with significant setup times, reconfiguration may require shutting down equipment for extended periods of time, causing significant losses in production. In cases where there is little flexibility and the ability to recover is limited, the problem-solving can be expensive and lengthy. Examples of such environments are often found in the chemical process industries, where tightly integrated equipment, limited intermediate storage space, significant setup times and volatile intermediate products can combine to render reconfiguration prohibitively costly in terms of lost output alone. Unfortunately, in many manufacturing environments most of the costs of reconfiguration and problem-solving remain hidden from shop personnel, since these may involve changes and disruptions in other departments and customer sites throughout the supply chain. Another important difficulty in complex multistage manufacturing systems is the fact that the effects of a disruption on system output may only be seen after a significant amount of time has elapsed, making it very difficult to link the disruption to its consequences.

To conclude, it is difficult to make a clear statement about the nature of the problem unless we can define the types of disruptions, describe the context, and define the actions that can be taken in the face of the disruptions. There is broad evidence in the literature that different types of reconfiguration actions are appropriate for different types of disruptions. In the following section we will give a brief overview of existing literature in the domain of scheduling with executional uncertainties. We will then conclude the paper with a discussion of some directions for future research and some of the weaknesses in the existing paradigms.

4. Existing research on scheduling with uncertainties

Over the last two decades a significant volume of research on the issues of scheduling with executional uncertainties has begun to emerge. We will review this research based on the problem formulation used: completely reactive approaches, robust scheduling approaches, and predictive–reactive scheduling. The latter of these is by far the most studied, and we therefore examine a number of specific issues related to this approach in more detail. We then proceed in the next section to examine issues associated with scheduling fully automated systems in the face of uncertainties, which has some interesting differences from those considered in much of the literature where the presence of humans to remove deadlocks is assumed.

4.1. Completely reactive approaches

This category of modelling approaches does not take any of the cause, context, impact, or inclusion issues into consideration per se. When looking for uncertainty principles, there is a void. The work is scheduled for the immediate future using normative information and assumptions and then nature takes over for the execution. These completely reactive approaches are characterized by least commitment strategies such as real-time dispatching that create partial schedules based on local information. Dispatching (Bhaskaran and Pinedo, 1991; Haupt, 1989; Holthaus and Rajendran, 2000; Ramasesh, 1990) examines the jobs currently available at the machine in question, and sometimes in its immediate environs. The next job to be processed is selected from among these by sorting and filtering them according to predefined criteria, and selecting the job at the head of the resulting list. This approach has many practical advantages. Its computational burden is in general extremely low, and the rules are usually intuitive and easy to explain to users. Empirical evidence from both industry and academia (e.g., Ovacik and Uzsoy, 1997) has repeatedly shown that for complex systems with high competition for capacity at key resources and relatively low uncertainty, global scheduling has the potential to significantly improve shop performance compared to localized or myopic dispatching. However, it should be noted that although these approaches do not exactly build a schedule, they do consider information from higher-level production plans through parameters such as due dates. A number of the more sophisticated dispatching procedures can invoke complex rules that allow them to consider the state of the system, at several different machines, and to take conditional actions based on this state. This type of policy has been extensively implemented in the semiconductor industry (Mohan and Clancy, 1990; Golovin, 1989).

A natural extension of the dispatching approach is to allow the system to select dispatching rules dynamically as the state of the shop changes. Early work in this area is that of Wu and Wysk (1989), who examine the problem of dispatching rule selection in a flexible manufacturing system environment. They divide the time horizon into shorter intervals. At the beginning of each interval a variety of dispatching rules are simulated, and the rule that yields the best performance is implemented for the next time period. A number of other authors have followed this approach and extended it in various ways, e.g., Kim and Kim (1994) and Jeong and Kim (1998). An extensive literature has evolved on the use of machine learning to select dispatching rules based on the state of the system. This literature is reviewed by Aytug et al. (1994a). One example of this work is by Piramuthu et al. (1991), who first use a simulation model of the manufacturing system under study to develop a characterization of how different dispatching rules perform in the system under different operating conditions. They then apply an inductive learning algorithm to this data to develop a decision tree that selects a dispatching rule whenever a significant change in system state is identified. Chen and Yih (1996) use a neural network to predict the dispatching rule to use under a certain system state. Aytug et al. (1994b) use genetic learning to select a population of rules for a given system configuration.

Another extension of these completely reactive approaches are those based on a number of independent, intelligent agents each trying to optimize its own objective function, which may differ from those of other agents. A number of researchers (e.g., Lin and Solberg, 1992; Duffie and Piper, 1987) have advocated scheduling systems of this nature, where a bidding mechanism is used to resolve conflicts between different agents. While the analogy to free-market economics is interesting, this approach still requires that the objectives of the individual agents be set in a manner that will ensure good overall system performance, which is not clear how to do. In addition, most of these approaches have been tested in the context of flexible manufacturing systems with relatively few machines.

4.2. Robust scheduling approaches

The next level of sophistication is shown in the second grouping of research results. In this grouping, the machine availability (cause—machine not ready) is modelled in some fashion using other context-free assumptions and with the only direct impact being machine availability to execute work. The robust scheduling approaches focus on creating a schedule which, when implemented, minimizes the effect of disruptions on the primary performance measure of the schedule. One way in which this is done (Daniels and Carrillo, 1997; Daniels and Kouvelis, 1995; Kouvelis and Yu, 1997; Kouvelis et al., 2000) is to consider a range of scenarios representing the results of different problem realizations (in this context, different realizations of disruptions to the schedule). A solution is then developed that optimizes performance under the worst possible scenario, with the objective of developing a schedule that will perform relatively well under a wide range of possible problem realizations. Several studies show that this approach often leads to significant improvements without degrading the expected performance over all scenarios significantly.

A second approach (Leon et al., 1993; Wu et al., 1999) is to minimize the expected degradation in performance measure, where the degradation is measured as the difference in objective function value between the predictive and realized schedules. Leon et al. (1993) also include a number of reconfiguration-related costs, such as the cost of changes in the start times and the cost of sequence changes. These approaches do not explicitly consider execution issues, since the formulation accounts for the fact that there will be disruptions prior to the execution of the schedule. The issue of predictability and graceful transition from a current system state is thus not considered. The implicit assumption is that the predictive schedule will be executed as is, at least as far as is feasible.

Taking a different approach, Mehta and Uzsoy (1998, 1999) and O'Donovan et al. (1999), develop predictive schedules to maximize the predictability of the realized schedule in both single machine and job shop environments subject to machine failures for a given rescheduling method. The former authors consider the primary performance measure of maximum lateness, while the latter consider the total tardiness. This is accomplished by estimating the effects of machine failures on the schedule and increasing the estimated job completion times by this amount to ensure that predicted job completion times are accurate estimates of those realized as execution proceeds. Results consistently show that significant benefits in schedule predictability can be obtained with minimal degradation of the primary performance measure. Bollapragada and Sadeh (1996) apply this approach to the job shop scheduling problem with total earliness–tardiness as performance measure. McKay et al. (2000) introduce a dynamic rescheduling approach that sub-optimizes for a period of time to allow the manufacturing situation a chance to re-stabilize and then progressively optimizes. Singer (2000) applies this idea to minimizing total tardiness in a job shop with uncertain processing times, obtaining similar results.

These approaches can be viewed as a form of under-capacity scheduling, where the amount of work scheduled in a time period is based on the historical performance of the equipment. Yellig and Mackulak (1997) provide an alternative formulation of this approach motivated by the control-theoretic models of Gershwin and his coworkers (e.g., Kimemia and Gershwin, 1983) Ashby and Uzsoy (1995) illustrate the performance of a particular under-capacity scheduling scheme in the face of uncertain arrivals of such orders. Horiguchi et al. (2001) illustrate the same ideas in the context of production planning in a semiconductor manufacturing facility.

4.3. Predictive–reactive scheduling

In predictive–reactive scheduling, scheduling is presented as a two step process. First, a predictive schedule representing the desired behavior of the shop floor over the time horizon considered, is generated. This schedule is then modified during execution in response to unexpected disruptions. The schedule actually executed on the shop floor after these modifications is called the realized schedule. The two main questions are when to initiate a rescheduling action and what that rescheduling action should be. Hence our discussion in this section will begin by examining the issue of when to initiate a rescheduling activity. We shall then discuss different formulations of the problem faced when a rescheduling action has been decided upon.

4.3.1. When to reschedule?

Regarding the first question, when to reschedule, the basic question that needs to be answered is when a disruption or an event has sufficient potential impact that a new schedule must be generated or some more localized remedial action taken. Church and Uzsoy (1992) provide a rough taxonomy of existing approaches beginning with two extremes. *Continuous rescheduling* approaches take rescheduling action each time an event that is recognized by the system, such as the arrival of a new job, occurs. *Periodic rescheduling*, on the other hand, defines a basic time interval T between rescheduling actions during which rescheduling actions are not permitted. Rescheduling actions are taken at times kT, where k is an integer. These points in time where rescheduling may be performed are referred to as rescheduling points. Any events occurring between rescheduling points are ignored until the following rescheduling point. Finally, they define *event-driven rescheduling*, in which a rescheduling action can be initiated upon the recognition of an event with potential to cause significant disruption to the system. Both continuous and periodic rescheduling can be viewed as special cases of event-driven rescheduling.

Clearly, continuous rescheduling runs the risk of initiating rescheduling activity in the face of events that do not cause significant disruption, expending computational resources and potentially causing unnecessary changes in the schedule with associated ill effects on the shop floor. The obvious drawback of periodic rescheduling is that it ignores events occurring between rescheduling points, which in an extreme case may render the current schedule impossible to execute, and in less serious situations runs the risk of yielding poor schedules. Hence a combination of the periodic and event-driven approaches appears attractive, in which a periodic rescheduling approach is implemented, but rescheduling activity can be invoked between rescheduling points if a disruption that is deemed sufficiently serious is observed. This latter approach is also commonly observed in practice, where schedules are often developed for some base horizon, such as a day or a shift, but are modified as needed during that period.

A number of authors have adopted the periodic and event-driven view of rescheduling and analyzed different approaches in this area. Church and Uzsoy (1992) consider the problem of minimizing maximum lateness on single-stage production systems involving single and parallel machines, where the only source of uncertainty is random job arrivals. They develop worst-case error bounds for the periodic approach assuming that an optimal algorithm is used to schedule the jobs available at each scheduling point. They then explore the performance of a combined periodic and event-driven approach, where additional rescheduling beyond what takes place at the rescheduling points can be caused by the arrival of a job with a tight due date. The basic insight obtained are summarized in Fig. 1, which plots the solution quality as a function of the number of rescheduling actions initiated. The underlying period of the periodic rescheduling policies

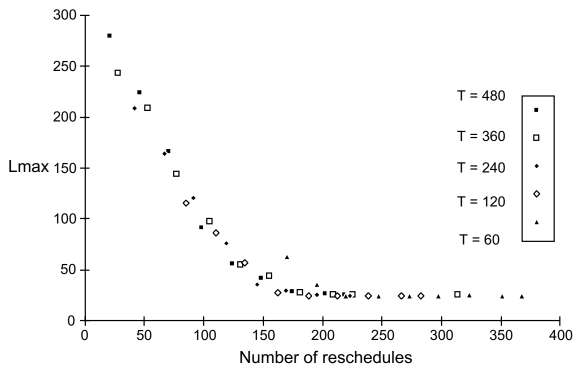

Fig. 1. No. of rescheduling events versus schedule performance—from Church and Uzsoy (1992).

are also indicated. These results indicate that schedule quality initially improves quite rapidly with more frequent rescheduling, but after a certain point yields almost no further gains. This is intuitive, since once the frequency of rescheduling activity exceeds the frequency of disruptions to the system the rescheduling activity is merely causing nervousness without improving the schedule quality. Another way of putting this is that a periodic response may well be sufficient to deal with the disruptions faced by the system, and that rescheduling with every system state change may be counterproductive, at least from the stability point of view. In this figure it is possible to interpret the number of rescheduling activities as a surrogate for the disruption to the shop floor caused by the changes of schedule. These results have been supported by a number of subsequent researchers for a variety of shop environments, e.g. Sabuncuoglu and Bayiz (2000) for the classical job shop environment under mean tardiness and makespan performance measures; Muhlemann et al. (1982) for job shops with multiple identical machines at each station; Sabuncuoglu and Karabuk (1998) for flexible manufacturing system with uncertain job processing times and machine breakdowns; Perry and Uzsoy (1993) for semiconductor testing operations with machine failures; Shafaei and Brunn (1999a,b) for open shops. Fang and Xi (1997) apply this type of approach to minimizing makespan in a flexible manufacturing system with essentially similar results.

A variety of authors have developed rolling horizon procedures, which are basically periodic rescheduling policies under the above taxonomy. These have been developed using a variety of techniques for solving the scheduling problems at each rescheduling point. Singer (2001) uses a heuristic decomposition procedure based on the shifting bottleneck procedure developed by Pinedo and Singer (1999). Qi et al. (2000) propose a similar framework using a multipopulation genetic algorithm. It is interesting to note that this type of approach can be used in completely deterministic systems as well as in those where the future state is uncertain.

4.3.2. *Predictive–reactive scheduling versus completely reactive approaches*

A number of authors have examined the question of when a periodic or event-driven rescheduling policy based on a global view of the scheduling problem can perform better than a completely reactive dispatching approach. Yamamoto and Nof (1985) compare the effects of a fixed optimization-based schedule, an event-driven rescheduling approach and dispatching rules in a FMS environment. They find that in the systems under study, a fixed optimization-based schedule obtained by a branch and bound algorithm out performs myopic dispatching rules in the face of machine failures, and is in turn outperformed by the event-driven rescheduling approach. Hutchison and Khumawala (1990) examine this question in a flexible manufacturing system environment where the only uncertainty is due to job arrivals at the start of planning periods. They find that a periodic rescheduling policy based on an optimization formulation developed by Hutchison et al. (1991) outperforms dispatching, especially when there is routing flexibility. Wan (1995) shows that when processing times are random, a global scheduling algorithm may yield poorer solutions than a dispatching policy. He also illustrates the dangers of using the mean processing time in a situation where processing times are random variables with relatively high coefficient of variation (e.g., exponentially distributed where the coefficient of variation is equal to one).

An important paper in this area is that of Lawrence and Sewell (1997), who compare the performance of a global scheduling heuristic based on the shifting bottleneck algorithm of Adams et al. (1988) with myopic, completely reactive dispatching rules in the presence of uncertain job processing times. They demonstrate that as processing time uncertainty increases, the difference in performance between the global method and the dispatching rules becomes less significant. They conclude that in systems with high uncertainty, completely reactive algorithms can be used with relative confidence, and question the benefits of global scheduling procedures in general.

In contrast, Barua et al. (2001) propose another approach in which a global schedule for the factory is developed using a periodic rescheduling policy. This schedule is not implemented directly, but rather serves to provide a priority index for the jobs as execution unfolds. The jobs are dispatched at the machines based on their start times in the global schedule. Extensive simulation experiments show that under a variety of operating conditions, including processing time uncertainty and machine failures, this approach significantly outperforms myopic dispatching rules. However, as the level of uncertainty becomes high relative to the frequency of rescheduling, performance becomes deteriorates to a level comparable to that of myopic dispatching rules.

Honkomp et al. (1999) describe a simulator for semi-continuous and batch processing manufacturing environments that can accept deterministic schedules and simulate both a deterministic and a stochastic realization of the schedule. The stochastic version can also use rescheduling logic. Running two versions of the simulation the authors compare the performance and robustness of the schedules. Two metrics are used for comparison.

$P_B = \overline{OF}/OF_{DB}$ is a measure of how well the average objective function value of the stochastic simulation compared to the objective function of the best deterministic schedule.

$D_B = SD/|OF_{DB}|$ is the standard deviation of the replicas of stochastic version compared to best deterministic objective function. This is used as a measure of robustness. In simulations without rescheduling schedules with the best performance also had the best robustness which is somewhat counter intuitive. In cases with rescheduling, rescheduling strategy with no penalties (i.e., can reschedule anything in the future) or no rescheduling created the best performance. Again those that had the best performance had the best robustness.

Matsuura et al. (1993) provide an extensive study of a slightly different rescheduling policy. In their approach, called switching, a predictive schedule is developed on a periodic basis. However, if the realized schedule is deemed to have deviated sufficiently from the predictive one, the system switches to using a dispatching rule for the remainder of the period. This approach is contrasted with using the predictive schedule throughout the period (by right-shifting jobs when delays occur) and dispatching approaches. They focus on three different types of disruptions: rush order arrival, specification changes (which cause new operations to be added to a job, or existing operations to be deleted), and machine failures. Their results are quite insightful: they show that when the frequency of disruptions is low, the predictive/reactive approaches outperform the dispatching. Once the level of disruption reaches a certain level, however, the dispatching begins to perform better than the predictive/reactive approaches.

We believe that the answer to this debate lies in the results of Matsuura et al. (1993) and Lawrence and Sewell (1997), and is hinted at in the results of several other papers. In an environment with little uncertainty, predictive/reactive methods based on global information and optimization techniques are highly likely to yield better schedules than completely reactive dispatching procedures. However, once the variability in the system exceeds a certain level, which appears to be system-dependent, the global information on which the predictive/reactive approaches are based becomes invalid, causing them to generate poor schedules due to solving the wrong problem: the problem data they use does not correspond to the problem encountered on the shop floor.

Having agreed with Lawrence and Sewell (1997) thus far, however, we do not believe that this insight should push us to disregard work on predictive/reactive scheduling methods. First of all, when a manufacturing system is subject to high levels of uncertainty we would suggest that management's time and resources would be better spent on understanding the sources of this variability and working to reduce it, rather than developing sophisticated scheduling logic. In addition, there are many manufacturing systems in which the difference in performance that can be obtained from a sophisticated scheduling procedure over a dispatching rule is simply not worth the amount of trouble that would be required to implement the global system. On the other hand, in capital-intensive environments such as semiconductor manufacturing, which require hundreds of unit processes and complex machinery and product routings, improvements of even a few percentage points in performance measures such as the average lead time may be worth millions of dollars.

It is interesting to note that a number of researchers have attempted to use global schedules as a complement to dispatching rather than to replace them. We have already mentioned the work of Barua et al. (2001), in which a global schedule generated on a periodic basis is used as a priority index in a dispatching rule, that outperforms myopic rules that do not use global information under a wide range of operating conditions. Similarly, Roundy et al. (1991) propose a price directed approach to the scheduling problem comprised of a global scheduler and a real-time dispatching module. The performance measure that they use is the weighted tardiness. They derive costs that are associated with performing a job at a particular time from the global scheduler that is passed on to the dispatching module. When a machine becomes free, the local dispatcher runs a fast algorithm to determine the job to be processed next based on the costs given by the global scheduler. With increasing shop complexity, their method performed well compared to dispatching rules.

Taking a rather different approach, Wu et al. (1999) propose a decomposition approach for scheduling in a job shop environment in which an optimization model using global information on the shop is used to develop a partial ordering of jobs in a manner that reflects their relative priorities based on global considerations, but also leaves room for dynamic decisions to be made as the state of the factory evolves. This approach is somewhat related to that of Baptiste and Favrel (1993), in that it essentially provides the execution agent with a partial schedule that can yield a number of different schedules. The authors show that their approach performs well in terms of robustness, where robustness is defined as the amount of degradation in performance as processing time variation increases. A similar approach which uses an activity on node representation similar to the disjunctive graph representation mentioned below, and again aims at specifying a set of schedules from which the user or a real-time scheduling system can select an appropriate decision in real time is proposed by Billaut and Roubellat (1996), who discuss a commercial implementation of a system based on this approach but do not provide detailed computational experiments.

In these three groupings of research results, the scope of uncertainty is very limited and the taxonomy introduced in Section 2 can be applied. The uncertainty is considered random, context-free, and the assumption of independence is pervasive. The cause of uncertainty is often machine availability (breakdown and repair) or some stochastic aspect of processing time that makes the start and finish times variable. Uncertainty is included neither in the initial schedule generation nor in the reactive regeneration.

4.3.3. Problem representations

An obvious issue in predictive–reactive scheduling is that of assessing the impact of a given disruption on an existing schedule. This is important for two different reasons. On the one hand, we need to assess the impact of a disruption to determine whether a rescheduling action is necessary. Once a rescheduling action has been decided upon, we may need an estimate of the impact of the disruption to select and execute an appropriate rescheduling action.

A number of authors study how to estimate the impact of a disruption at a particular point in the execution of a predefined schedule (e.g., Abumaizar and Svestka, 1997; Wu et al., 1999; Mehta and Uzsoy, 1998; Wu and Li, 1995). It is interesting to note that many of these papers use algorithms that can be inferred from the well-known disjunctive graph representation of the job shop scheduling problem, although not all of them appear to be aware of this. In this representation a general multimachine scheduling problem can be elegantly represented using a disjunctive graph $G(N, A, E)$ (Roy and Sussmann, 1964). Each node in the node set N corresponds to an operation to be processed on some machine, with the addition of sink, source and job completion nodes. The source node 0 has conjunctive (directed) arcs of length zero emanating from it to the first operation of each job. Each completion node j^* for job j has a conjunctive arc incident into it from the last operation of job j and a conjunctive arc incident from it to the sink node F. The remaining conjunctive arcs in the set A represent precedence relations between the operations of the individual jobs. Each pair of disjunctive arcs in the set E captures the relationship between operations to be processed on the same machine, and consists of a pair of arcs with opposite orientations, at most one of which can appear in any path in the graph. The set E forms m cliques (complete sub-graphs) of disjunctive pairs, one clique for each of the m machines. All operations belonging to a clique have to be processed on the same machine and thus cannot overlap in time. Hence a scheduling decision corresponds to fixing the disjunctive pair of arcs in one of the two possible orientations, i.e., deciding which of the two operations represented by the nodes will be scheduled before the other. All arcs incident from a node representing an operation have length equal to the processing time of that operation. A number of authors have discussed at length how to adapt this representation to a range of different shop conditions and performance measures (Ovacik and Uzsoy, 1997). However, once this graph has been constructed, the effect on operation start and end times can be calculated directly using longest path calculations in this graph, by updating the duration of the operation during whose processing the disruption occurs. Several of the algorithms mentioned above, notably those of Abumaizar and Svestka (1997) and Li et al. (1993) essentially develop these ideas independently.

It is interesting to observe that there are no representations of which we are aware that explicitly model the uncertainty in the representation, or perform any but the simplest calculations based on uncertainty. One would think that stochastic project scheduling methods such as the project evaluation and review technique (PERT), which have been well-studied for decades, would have interesting insights for scheduling researchers in this area. However, this literature does not seem to have had much effect on the research reviewed in this paper, probably because the underlying mathematics is much closer to pure stochastic scheduling that deterministic scheduling, and hence many researchers familiar with deterministic scheduling approaches are not comfortable with these approaches.

4.3.4. Formulations of the rescheduling problem

Given that rescheduling will be carried out, researchers have examined different formulations of the scheduling problem encountered at a specific rescheduling point. Many of these approaches consider both a primary measure of schedule performance that must be maintained, as well as some measure of the disruption caused by rescheduling the shop. This approach automatically leads to the formulation of the rescheduling problem as a multiobjective scheduling problem, where the issue is to develop schedules that are satisfactory with respect to both sets of criteria.

Naturally, the approaches to this issue have followed the standard approaches to multiobjective problems: hierarchical approaches, in which one performance measure is selected as more important than the other; weighted sums of the different objectives, and generation of all sets of efficient (Pareto-optimal) schedules. Unal et al. (1997) consider a static rescheduling problem in which a number of new jobs must be inserted into an existing schedule so as to minimize the total completion time of the new jobs without causing existing jobs to miss their deadlines. Hence the due date performance of existing jobs is considered a priority, and is imposed as a constraint on the secondary criterion of the total completion time of the new jobs. Alagoz and Azizoglu (2003) consider the problem of rescheduling a parallel machine workcenter subject to disruptions and provide heuristic algorithms to minimize the number of rescheduled jobs subject to optimizing the total completion time of all jobs in the system.

In the matchup scheduling approach (Bean et al., 1991; Akturk and Gorgulu, 1999), the objective is for the realized schedule to return to the predictive schedule within a certain time of the disruption occurring. This approach will clearly yield high-quality schedules if there is sufficient idle time in the original predictive schedules.

Leon et al. (1994) show that the rescheduling problem can be formulated as a stochastic control problem using decision trees. The authors formulate the problem as an N-step game where the objective is a convex combination of makespan and deviation from the original (or previous) predictive schedule. At each decision node (X_k) the controller can take one of N_a corrective actions in anticipation of a particular disruption (proactive) or because of a particular disruption (reactive). It is however important to note that the actions do not correspond to different heuristics but application of the same heuristic to different type of expectations or disruptions. The heuristic solves a one-machine scheduling problem ($P1$) in which the expected disruption is inserted as a job with known arrival time and duration. Once the new schedule is generated (node S_k) the system receives a disruption or a monitoring event that takes it to the next decision node (X_{k+1}). The authors describe methods to manage the size of the decision tree by sampling disruptions and available actions. The objective function value of a state V_k can be formulated as a recursive function of the future decisions and disruptions where V_N is the value to be minimized (solutions of $P1$ and V_N are given in two other papers by the authors) and the solution specifies a policy (a path from root to node). It is important to note that the decision tree constructed this way is a subset of that corresponding to the real phenomenon, so the solution to V_N is sub-optimal.

The formulation allows for modelling machine breakdowns, scheduled outages such as maintenance. It is also possible to incorporate monitoring epochs at scheduled time intervals. The controller is tested using simulation on various settings against total reschedule and right-shift policies (note that the controller has to compute an N-step policy based on what the simulation has just presented at step 1 (i.e., first disruption or first monitoring event)). As expected the quality of solution depends on N and monitoring frequency. In different machine reliability scenarios the controller outperforms other policies (note that by default the policy can include a right-shift action at a given node). The authors also demonstrate that the method is robust to the existence of estimation errors (i.e., disruption distributions) but it is sensitive to the initial off-line schedule (robust scheduling solutions seems to yield better results).

A considerable number of researchers have viewed the rescheduling problem as that of classifying the disruptions as they occur and selecting an appropriate rescheduling action from among a suite of options. Jain and Elmaraghy (1997) and Dutta (1990) suggest a number of essentially rule-based heuristic procedures that are invoked in the face of different events such as machine failure and the arrival of rush orders. One tool for this has been case-based reasoning (Koton, 1989). Dorn (1995) describes a case-based scheduler that creates a schedule using tabu-search (as a constraint satisfaction search algorithm) and that suggests repair strategies using its case base. Like all case-based systems it stores relevant system states and the solution used in this state. It can repair its case-base if existing cases either fail to match a system state or produce bad results. O'Kane (2000) describes a knowledge-based system for scheduling in FMSs that learns from simulation traces. The system logs the disruption states and the decisions made through out a simulation run. At the end of each simulation run the outcome is assessed and the knowledge base is updated if necessary. The system state is monitored by attributes like "number of interruptions encountered until now", "percentage of schedule completed", "source of interruption", etc. and some of the actions suggested are "abort schedule", "re-route pallets", "flag for more preventive maintenance", etc. However, no experimental results or details of the learning algorithm are reported.

Miyashita and Sycara (1995) describe a case-based system that is able to create an "optimized" schedule that is in line with the scheduler's preferential knowledge (or objective function). The system starts with an initial schedule and compares the performance to that of the scheduler's. If the current performance is unacceptable the system "repairs" the schedule until it meets the required criteria. During the repair process the system randomly identifies an activity to be repaired and a case matching the current state is invoked. The action suggested by the case is taken and the result is evaluated. If the result is a success the system chooses the next activity to repair, otherwise another case that match the current state is invoked. The system has the ability to store both the failures and successes for a case. Due to its "repair" centric vision the system can be used both for predictive and reactive scheduling. In case of a disruption the system first resolves any infeasibility with a right-shift and then starts the repair cycle to meet the user's preferences. Experiments on reactive scheduling show that CABINS outperforms a constraint based search method (complete rescheduling). Miyashita (2000) discusses how reinforcement learning can be incorporated for case acquisition (search control knowledge) in a case-based scheduler. The need for human interaction during training is eliminated since the system is able to replace the concept of acceptable and unacceptable with actual rewards (objective function values). Through trial and error the system can determine what actions yield acceptable results (high reward) and which do not. However, comparisons with CABINS (Miyashita and Sycara, 1995) reveal that the cases acquired this way are inferior.

Szelke and Markus (1995) describe a case-based system as in Dorn (1995). The knowledge of a scheduler is abstracted at four levels (1) state recognition, (2) policy selection, (3) policy implementation and (4) policy execution. The blackboard control architecture enables the system to carry out different tasks at different levels of abstraction in parallel following the goal/plan/task/action structure. The controlling unit of the blackboard ensures that conflicting plans are resolved and system feasibility is maintained for a time window. The necessary knowledge to create and repair schedules is saved as cases and rules that are instantiated by the control unit for the appropriate task at hand. Another basic approach to scheduling in the face of uncertainty in the artificial intelligence community (e.g., Monostori et al., 1998; Smith, 1993; Szelke and Kerr, 1994) on scheduling in the face of disruptions. One basic approach has been to formulate the scheduling problem as a constraint satisfaction problem, in which the objective is to find a feasible solution. They then use a variety of heuristic search techniques to "repair" schedules, i.e., to reschedule jobs to restore feasibility once infeasibilities have been detected. Once infeasibility has been accomplished, they may also attempt to seek for schedules with good performance. Examples are Sadeh et al. (1993) and Zweben et al. (1993, 1994), who accomplish this by using simulated annealing, while Smith and his co-workers use constraint-guided heuristic search (Hasle and Smith, 1994; Henseler, 1995; Smith et al., 1990a,b; Smith, 1994). The rescheduling research is also limited in the causal and impact dimensions. With the exception of O'Donovan et al. (1999) and McKay et al. (2000), the rescheduling work is also context-free.

In summary, the majority of recent research efforts typified by the three groupings address a very small number of uncertainty causes, and implicitly assume context-free situations and independence. The research also restricts the impact to time. The rescheduling research attempts to include uncertainty concepts directly (e.g., performance objectives) but there is little work on predictive generation of schedules that anticipate uncertainty and the impacts.

4.3.5. Implementation and bridging issues

Another body of work has addressed the issues of how to implement systems in which rescheduling activities must take place and how to bridge theory and practice. At the theoretical level McKay and Wiers (1999) argue that uncertainty and its intricacies must be addressed before theory can be readily used in practice. For what can be described as severely restricted situations, several authors have proposed system architectures, describing how automated shop floor control and factory planning systems can be implemented. A common thread running through this focused work is the idea of separating the scheduling function into a planner and a dispatcher, where the planner develops a predictive schedule, and the dispatcher executes it to the best extent possible given the current situation on the shop floor. Most of these architectures also require a monitoring capability that will determine when the realized schedule on the shop floor has deviated sufficiently from the predictive schedule that a rescheduling action of some kind is required. Bauer et al. (1991), Grant and Nof (1989), Roundy et al. (1991), Smith et al. (1990a,b) all propose variations of this approach. Henning and Cerda (2000) discuss these issues in the context of process industries. Chang and Luh (1997) propose an approach for integrating cell-level controllers with global scheduling algorithms. In their approach a global schedule is generated for a longer time horizon, and rescheduling may take place for shorter time horizons. Finally, a dispatching module executes the current schedule. Du and Chiou (2000) present an approach based on version management in an object oriented database to implement an event-based rescheduling scheme.

4.3.6. Summary

These five categories of scheduling research concentrate on sequence generation or regeneration driven by localized objectives. This is only part of the problem when considering schedule execution under uncertainty. The research does not consider the many interrelationships that exist in real situations between jobs, between machines, and between processes. These interrelationships become important when uncertainty exists and critical events occur. For example, a machine goes down and affects a specific job. This job's remaining operations are scheduled in the future and this can affect other jobs and their operations competing for the resource. What does one do with the crashed job and its remaining pieces? How can other jobs be dynamically re-routed to choose alternative processes or resources? In a non-automated factory situation, the humans perform the problem-solving and do a variety of things to create new capability, juggle requirements, and keep the manufacturing system flowing. The batch sizes may be changed, equipment re-wired, old processes dusted off, people cajoled to perform different tasks, and so forth. The schedule is executed—there is product being made. The human provides the knowledge and skill to reconfigure the problem and create the necessary solutions. How does this work in an automated factory? In a system with real-time control, machines and processes may still fail and create uncertainty. Within black-box manufacturing, the automatic schedule generation and regeneration can ignore the impacts of uncertainty or incorporate the ability to deal with it. The following section discusses uncertainty in automated settings and reviews current research.

5. Execution under uncertainty in automated settings

In theory, a highly automated system is devoid of the context-sensitive attributes discussed in Section 3 and is suitable for consideration for mathematical approaches to scheduling (e.g., Wiers, 1997). Furthermore, an automated system is usually cushioned and protected from many of the sources of uncertainty that face general manufacturing. There is also reduced uncertainty in the process since the system is automated and has to be specified in detail. These characteristics reduce the overall challenge associated with uncertainty. This is not to say that uncertainty is absent or minimized. The yield may be uncertain, creating uncertainty in batch sizes and affecting start and finish times. Machines and material handling equipment can fail. Prototype work put through the system can destabilize existing processes, requiring the system to be retuned. Prototype work has also been known to damage equipment. Material used in the process is also subject to variability. Hence, there can be enormous quantities of uncertainty in an automated system—depending on the system.

An automated system that does not deal proactively with uncertainty stalls or can enter chaotic behavior. Jobs can block other jobs, resources can be in conflict, causing the whole system to shut down. A progressive system would be able to handle many of the common forms of uncertainty and react appropriately. Not all forms of uncertainty can be predicted or corrected for, but it is reasonable to expect that situations like having one machine out of a group of similar machines fail would not bring the system to a halt. That is, the system must have its operational feasibility maintained—without the help of human problem-solving and intervention. Specifically, whatever control mechanism is used, it must allocate resources so that if some critical resource fails, the failure does not propagate through and cause major system disruption. Indeed, the system should continue to operate smoothly and autonomously while the failed resource is being repaired or replaced. This requires allocating resources so that jobs requiring a failed resource do not block others that do not require the failed resource. It also requires making full use of potential process redundancies and flexible routing capabilities. These issues are important for highly automated facilities which large capitalization investments. For example, robust operation is particularly important in the semiconductor industry, where the estimated annual cost of unplanned downtime is currently in the range of $1 billion (Wohlwend et al., 1996).

An area of automated manufacturing research that focuses on operational feasibility and robustness is structural control. Structural control has several objectives, one key criteria relevant to the discussion on uncertainty is its goal to prevent any allocation of system resources that adversely affects its ability to continue production. Because the control mechanisms operate in real-time, there are numerical complexity issues relating to configurability analysis and resource conflict resolution (e.g., Lawley et al., 1997; Lawley and Reveliotis, 2001). There are complementary concepts to structural control that impact an automated system's ability to handle uncertainty. For example, the ability to handle the majority of possible disruptions might require that all tools have multiple copies or that all routes be acyclic. These types of special manufacturing structures can have important implications for system design and have been applied to allocating machine capacity and tooling in automated manufacturing systems (Lawley and Reveliotis, 2001; Gebraeel and Lawley, 2001).

It is clear that the objective of structural control may conflict with that of the scheduling logic and the predictive schedule. Indeed, unless scheduling techniques account for the resource allocation logic in the structural controller, the predictive schedule is almost certain to be infeasible with respect to the structural controller. These conflicts must be resolved in favor of the structural controller, since to do otherwise is to invite a catastrophic system failure that devastates system performance. Integrating structural control logic into the predictive scheduling process is not feasible due to the combinatorial explosion that would result. Thus, a major research direction is to integrate predictive–reactive scheduling with structural control in automated systems. This will be particularly important in the coming 300 mm wafer generation in the semiconductor industry, where the use of manufacturing automation is expected to rise to over 95% (Wohlwend et al., 1996).

It is interesting to note that with the exception of McKay et al. (1995), the scheduling literature considered in the previous section essentially ignores the need for structural control of any kind. The style of structural control found in McKay et al. was human centered and did not address the specific challenges found in highly automated plants when the control has to be done by software and precise calculations. The structural control problems faced by automated systems have been ignored completely (i.e., the numerical complexity, dynamic restructuring of the problem, and extreme constraint relaxation). Note that the structural control policy in use may actually become an additional source of uncertainties for the scheduling system, since the structural control policy in force may override scheduling decisions that may jeopardize successful system operation. However, the perspective of not allowing the system to enter "undesirable" states is useful for considering the problem of executing a schedule.

6. Discussion and future directions

In this section we will briefly summarize our conclusions from the discussion above, and suggest a number of broad areas for future research.

6.1. Problem formulation

The vast majority of the scheduling research we are aware of does not explicitly consider execution issues such as uncertainty, but implicitly assumes that the global schedule will be executed exactly as it emerges from the algorithm generating it. The theory does not address different causes, the context in which uncertainty arises, or the various impacts that might result (McKay and Wiers, 2001). Conceptually, extending the scheduling model to include all essential constraints will permit direct execution of the global schedule. However, the inclusion of additional constraints into global scheduling models significantly increases the complexity, and therefore, the computational burden, of both the schedule generation and rescheduling tasks, which are in general NP-hard to begin with.

In terms of formulations, the vast majority of the current literature falls into the category of predictive–reactive scheduling, where a predictive schedule is released to the shop floor and then progressively modified to allow it to function effectively as disruptions occur. In effect, each time a disruption occurs and is accommodated a new predictive schedule emerges that remains in force until the next disruption. A central theme of this research is that of schedule repair—the need to have a schedule in existence that is feasible at all times. Clearly, this requires the constant monitoring of the status of the shop floor against the predictive schedule, and possibly the interruption of processing on the shop floor while the new schedules are generated. The research is focused upon the machine failure as the cause and time as the only impact.

It is noteworthy that in this area many different researchers appear to have trodden essentially the same ground with essentially the same results. The two main conclusions seem to be (i) that rescheduling more frequently does not make things worse, but does not make things better either beyond a certain frequency, and (ii) if the level of uncertainty is low enough, an optimization-based predictive scheduling algorithm can outperform an on-line, dispatching algorithm but the converse is true once uncertainty exceeds a certain threshold. We would suggest that efforts to quantify these thresholds and relate them to system parameters would be a useful direction for future work, since at present we have little understanding of how the behavior of these thresholds changes with the different kinds of uncertainty present in the system. In addition, it is difficult to extract general insights from the current literature beyond the two broad conclusions stated above—much of the work is simulation based, and hence must be interpreted in the context of the specific system configurations studied—most papers examine only one basic system structure. Studies that simply reiterate the two broad conclusions reached above for different system topologies are of limited value.

Another point to be made in passing is a methodological one. Many papers present what are essentially heuristic algorithms for an optimization formulation of the rescheduling problem, but often give only an illustrative example to show how the procedure works. In order to gain an in-depth understanding of the performance of any heuristic under different conditions it is essential to conduct well-designed computational experiments and analyze their results in an appropriate manner. Rardin and Uzsoy (2001) discuss some of these methodological issues in depth.

It may well be worthwhile to consider the predictive schedule in a somewhat different role—that of providing a guideline, or information, on the relative priority of jobs based on overall factory status—as relating to the various purposes of scheduling outlined in Section 1. Viewed in this manner, the global schedule does not even need to be feasible at all times—it just needs to capture a global picture of resource contention and give relative priorities to jobs. The actual issue of which job goes next on which machine can be handled by a dispatching-like system which considers the position of the job in the global schedule in addition to current shop-floor status. Hence, the global scheduler can be viewed as complementing and extending, not replacing, existing real-time dispatch systems. This is consistent with the hierarchical approach to production control adopted by many researchers over the last several decades, and also corresponds closely to industrial practice in a wide range of industries.

The formulation can also be extended to address areas of uncertainty identified by the taxonomy introduced in Section 2. It is clear that the existing research is extremely narrow and that our current research and models do not capture the variety of uncertainty encountered in real situations. We need to consider the inclusion of context and impact if we are to be able to model what is going to happen and to derive suitable solutions to the problem.

6.2. Estimation of reconfiguration costs

Much of the literature has focused exclusively on schedule performance, and has ignored the costs of reconfiguration, although an increasing body of work is now beginning to include these in a variety of forms. However, there is clearly no broad agreement on what the best way of modelling such costs is. We conjecture that these costs are likely to be system-dependent, and should probably take into account at least some of the repercussions at other nodes of the supply chain that will be forced to reconfigure to some degree by changes in the production schedule at the current shop. A systematic approach to the estimation of different costs of reconfiguration would be of considerable theoretical and practical interest.

Another issue to consider here is that of how to evaluate the performance of systems in the face of disruptions. The standard practice is to use long-run steady state performance measures in simulation studies, but this may well miss crucial dynamic aspects of system behavior. For example, Uzsoy et al. (1993) considered the performance of different dispatching rules in a shop with processing time uncertainty and time-varying job arrival rates. Since they used long-run steady state statistics to compare the algorithms, much of this variation had little apparent effect on system performance. However, shop-floor personnel do not manage in long-run, but over short time periods such as shifts or weeks. Much of their behavior is determined by the considerations of the effects of their actions over this time frame, which is not captured by long-run statistics.

6.3. Using available information on the nature of disruptions

Another interesting aspect of much of the predictive–reactive scheduling research is its implicit assumption that we know absolutely nothing about disruptions that will allow us to take some action while building the predictive schedule to mitigate their effects. In practice, there is often statistical information on at least some kinds of disruptions, such as machine failures, which can be employed to develop predictive schedules capable of surviving disruptions. In some industries, such as semiconductor manufacturing, it is often possible to assess the state of a machine's health and assign work accordingly, due to the various monitoring capabilities available. Mehta and Uzsoy (1998) give one example of how this information can be used, as does the robust scheduling approach of Daniels and Kouvelis (1995). However, this area is clearly worthy of more study. Similarly, different jobs and machines often respond in different ways to disruptions, and an experienced human scheduler will often exploit such knowledge in developing and reconfiguring production schedules. O'Donovan et al. (1999) illustrate one way of incorporating this type of information into scheduling heuristics. The latter paper combines the two ideas of using statistical information on disruptions in developing the predictive schedule and in rescheduling after the disruption has occurred. McKay et al. (2000) and Black (2001) also discuss how additional information can be used and illustrate how to create hybrid scheduling heuristics to incorporate such knowledge. The results of this small body of work are very promising, indicating that significant reductions in reconfiguration-related costs can be obtained with very minor sacrifices from the conventional scheduling performance measures.

6.4. Integration with structural control

A surprising gap in the literature is the almost total lack of connection between the extensive literature on structural control of automated manufacturing systems and scheduling with disruptions. The two bodies of work differ quite fundamentally in their approach to the problem. The scheduling literature tends to view the problem as that of updating a plan in the face of unexpected changes in the execution environment. The emphasis in this literature is on optimizing shop performance, exactly or approximately, over a period of time, where the schedule performance is generally defined in terms of the movement of jobs through the entire shop—due date performance or flow times, for instance. In contrast, structural control focuses on a very short time frame, with the goal of preventing the system from entering a state that may lead to catastrophic failure. There is no consideration of managerial priorities such as due dates.

However, it is clear that the two functions interact substantially. An improved perspective integrating these two viewpoints would be very useful in practice.

7. Conclusion

We started with a broader definition of what scheduling is and why scheduling is performed which allowed a fuller discussion on uncertainty. A four-dimensional taxonomy for uncertainty was introduced that was then used to frame a number of research areas. The literature review and discussion clearly indicates that while some limited work and progress has been made in the area, much remains to be done. For too long the research community has failed to either understand or appreciate what scheduling is in a real situation and what the key dimensions are—uncertainty is one such dimension.

Acknowledgements

This research was supported by Consilium of Applied Materials and the National Science Foundation under Grant No. DMI-0085047. Part of the research was supported by NSERC grant OGP 0121274 on 'Adaptive Production Control'.

References

Abumaizar, R.J., Svestka, J.A., 1997. Rescheduling job shops under random disruptions. International Journal of Production Research 35, 2065–2082.

Adams, J., Balas, E., Zawack, D., 1988. The shifting bottleneck procedure for job shop scheduling. Management Science 3, 391–401.

Akturk, S.M., Gorgulu, E., 1999. Match-up scheduling under a machine breakdown. European Journal of Operational Research 112 (1), 81–97.

Alagoz, O., Azizoglu, M., 2003. Rescheduling of identical parallel machines under machine eligibility constraints. European Journal of Operational Research 149 (3), 523–532.

Ashby, J.R., Uzsoy, R., 1995. Scheduling and order release in a single-stage production system. Journal of Manufacturing Systems 14, 290–306.

Aytug, H., Bhattacharyya, S., Koehler, G.J., Snowdon, J.L., 1994a. A review of machine learning in scheduling. IEEE Transactions on Engineering Management 41 (2), 165–171.

Aytug, H., Koehler, G.J., Snowdon, J.L., 1994b. Genetic learning of dynamic scheduling within a simulation environment. Computers and Operations Research 21, 909–925.

Baptiste, P., Favrel, J., 1993. Taking into account the rescheduling problem during the scheduling phase. Production Planning and Control 4, 349–360.

Barua, A., Narasimhan, R., Uzsoy, R., 2001. Implementing global factory schedules in the face of stochastic disruptions. Research Report, School of Industrial Engineering, Purdue University.

Bauer, A., Bowden, R., Browne, J., Duggan, J., Lyons, G., 1991. Shop Floor Control Systems: From Design to Implementation. Chapman and Hall, London.

Bean, J.C., Birge, J.R., Mittenhal, J., Noon, C.E., 1991. Matchup scheduling with multiple resources, release dates and disruptions. Operations Research 39, 470–483.

Bhaskaran, K., Pinedo, M., 1991. Dispatching. In: Salvendy, G. (Ed.), Handbook of Industrial Engineering. John Wiley, New York. Chapter 83.

Billaut, J.C., Roubellat, F., 1996. A new method for workshop real time scheduling. International Journal of Production Research 34, 1555–1579.

Black, G., 2001. Predictive, Stochastic, and Dynamic Extensions to Aversion Dynamics Scheduling. Ph.D. Thesis, University of Alabama, Huntsville.

Bollapragada, R., Sadeh, N., 1996. Cost-based approaches to stochastic job shop scheduling problems. Research Report, Robotics Institute, Carnegie-Mellon University.

Chang, J.W., Luh, Y.P., 1997. Integration of scheduling and control in a job shop. Journal of the Chinese Institute of Engineers 20, 67–76.

Chen, C., Yih, Y., 1996. Identifying attributes for knowledge base development in dynamic scheduling environments. International Journal of Production Research 34 (6), 1739–1755.

Chen, F., Drezner, Z., Ryan, J.K., Simchi-Levi, D., 2000. Quantifying the bullwhip effect in a simple supply chain: The impact of forecasting, leadtimes and information. Management Science 46, 436–443.

Church, L.K., Uzsoy, R., 1992. Analysis of periodic and event-driven rescheduling policies in dynamic shops. International Journal of Computer Integrated Manufacturing 5, 153–163.

Coburn, F.G., 1918. Scheduling: The coordination of effort. In: Mayer, I. (Ed.), Organizing for Production and Other Papers on Management 1912–1924. Hive Publishing, Easton. 1981, pp. 149–172.

Daniels, R.L., Carrillo, J.E., 1997. β-robust scheduling for single machine systems with uncertain processing times. IIE Transactions on Scheduling and Logistics 29, 977–985.

Daniels, R.L., Kouvelis, P., 1995. Robust scheduling to hedge against processing time uncertainty in single-stage production. Management Science 41, 363–376.

Dauzere-Peres, S., Lasserre, J.-B., 1994. An Integrated Approach in Production Planning and Scheduling. Lecture Notes in Economics and Mathematical Systems. Springer-Verlag, Berlin.

Dorn, J., 1995. Case-based reactive scheduling. In: Kerr, R., Szelke, E. (Eds.), Artificial Intelligence in Reactive Scheduling. Chapman and Hall, London, pp. 32–50.

Du, T.C., Chiou, R., 2000. Applying version management of object-oriented database technology in reactive scheduling. International Journal of Production Research 38, 1183–1200.

Duffie, N.A., Piper, R.S., 1987. Non-hierarchical control of a flexible manufacturing cell. Robotics and Computer-Integrated Manufacturing 3, 175–179.

Dutta, A., 1990. Reacting to scheduling exceptions in FMS environments. IIE Transactions 22, 300–314.

Emerson, H., 1913. Twelve Principles of Efficiency. The Engineering Magazine, New York.

Fang, J., Xi, Y., 1997. A rolling horizon job shop rescheduling strategy in the dynamic environment. International Journal of Advanced Manufacturing Technology 13, 227–232.

Gantt, H.L., 1919. Organizing for Work. Allen and Unwin, London.

Gary, K., Kempf, K.G., Uzsoy, R., Smith, S.F., 2000. Evaluation and comparison of production schedules. Computers in Industry 42, 203–220.

Gebraeel, N., Lawley, M., 2001. Deadlock detection, prevention, and avoidance in automated tool sharing systems. Research Report, School of Industrial Engineering, Purdue University.

Golovin, J., 1989. Real time dispatching for optimum scheduling. In: Proceedings of the A&D Symposium, APICS.

Grant, F.H., Nof, S.Y., 1989. Automatic adaptive scheduling of multiprocessor cells. Research Report, School of Industrial Engineering, Purdue University.

Harmonosky, C.M., Robohn, S.F., 1991. Real-time scheduling in computer-integrated manufacturing: A review of recent research. International Journal of Computer-Integrated Manufacturing 4 (6), 331–340.

Hasle, G., Smith, S.F., 1994. Directing an opportunistic scheduler: An empirical investigation on reactive scenarios. In: Kerr, R.M., Szelke, E. (Eds.), Preprints of Second International Workshop on Knowledge-Based Reactive Scheduling. Computer and Automation Research Institute, Hungarican Academy of Sciences, Budapest, pp. 7–17.

Haupt, R., 1989. A survey of priority rule-based scheduling. OR Spektrum 11, 3–16.

Henning, G.P., Cerda, J., 2000. Knowledge-based predictive and reactive scheduling in industrial environments. Computers and Chemical Engineering 24, 2315–2338.

Henseler, H., 1995. REAKTON: A system for event independent reactive scheduling. In: Kerr, R., Szelke, E. (Eds.), Artificial Intelligence in Reactive Scheduling. Chapman & Hall, London.

Holthaus, O., Rajendran, C., 2000. Efficient jobshop dispatching rules: Further developments. Production Planning and Control 11, 171–178.

Honkomp, S.J., Mockus, L., Reklaitis, G.V., 1999. A framework for schedule evaluation with processing uncertainty. Computers and Chemical Engineering 23, 595–609.

Horiguchi, K., Raghavan, N., Uzsoy, R., Venkateswaran, S., 2001. Finite capacity production planning algorithms for a wafer fabrication facility. International Journal of Production Research 39, 825–842.

Hutchison, J., Khumawala, B., 1990. Scheduling random flexible manufacturing systems with dynamic environments. Journal of Operations Management 9, 335–351.

Hutchison, J., Leong, K., Snyder, D., Ward, P., 1991. Scheduling approaches for random job shop flexible manufacturing systems. International Journal of Production Research 29, 1053–1067.

Jain, A.K., Elmaraghy, H.A., 1997. Production scheduling/rescheduling in flexible manufacturing. International Journal of Production Research 35, 281–309.

Jeong, K.C., Kim, Y.D., 1998. A real-time scheduling mechanism for a flexible manufacturing system using simulation and dispatching rules. International Journal of Production Research 36, 2609–2626.

Kim, M.H., Kim, Y.D., 1994. Simulation-based real time scheduling mechanism in a flexible manufacturing system. Journal of Manufacturing Systems 13, 85–93.

Kimemia, J., Gershwin, S.B., 1983. An algorithm for the computer control of a flexible manufacturing system. IIE Transactions 15, 353–362.

Koton, P., 1989. SMARTplan: A case-based resource allocation and scheduling system. In: Proceedings of DARPA Workshop on Case-Based Reasoning, Pensacola Beach, FL, pp. 285–289.

Kouvelis, P., Yu, G., 1997. Robust Discrete Optimization and Its Applications. Kluwer Academic Publishers.

Kouvelis, P., Daniels, R.L., Vairaktarakis, G., 2000. Robust scheduling of a two-machine flow shop with uncertain processing times. IIE Transactions on Scheduling and Logistics 32, 421–432.

Lawley, M., Reveliotis, S., 2001. Optimal deadlock avoidance in sequential resource allocation systems: Hard and easy cases. International Journal of Flexible Manufacturing Systems 13, 385–404.

Lawley, M., Reveliotis, S., Ferreira, P., 1997. Design guidelines for deadlock handling strategies in flexible manufacturing

systems. International Journal of Flexible Manufacturing Systems 9 (1), 5–30.
Lawrence, S.R., Sewell, S.E., 1997. Heuristic, optimal, static and dynamic schedules when processing times are uncertain. Journal of Operations Management 15, 71–82.
Leon, V.J., Wu, S.D., Storer, R.H., 1993. Robustness measures and robust scheduling for job shops. IIE Transactions 26, 32–43.
Leon, V.J., Wu, S.D., Storer, H., 1994. A game-theoretic control approach for job shops in the presence of disruptions. International Journal of Production Research 32, 1451–1476.
Li, R.-K., Shyu, Y.-T., Adiga, S., 1993. A heuristic rescheduling algorithm for computer-based production scheduling systems. International Journal of Production Research 31, 1815–1826.
Lin, G.Y.J., Solberg, J.J., 1992. Integrated shop-floor control using autonomous agents. IIE Transactions 24, 57–71.
Matsuura, H., Tsubone, H., Kanezashi, M., 1993. Sequencing, dispatching, and switching in a dynamic manufacturing environment. International Journal of Production Research 31, 1671–1688.
McKay, K.N., Wiers, V.C.S., 1999. Unifying the theory and practice of production scheduling. Journal of Manufacturing Systems 18 (4), 241–255.
McKay, K.N., Wiers, V.C.S., 2001. Decision support for production scheduling tasks in shops with much uncertainty and little autonomous flexibility. In: MacCarthy, B., Wilson, J. (Eds.), Human Performance in Planning and Scheduling. Taylor and Francis, New York, pp. 165–178.
McKay, K.N., Safayeni, F., Buzacott, J.A., 1995. An information systems based paradigm for decision making in rapidly changing industries. Control Engineering Practice 3 (1), 77–88.
McKay, K.N., Morton, T.E., Ramnath, P., Wang, J., 2000. Aversion dynamics—Scheduling when the system changes. Journal of Scheduling 3 (2), 71–88.
Mehta, S.V., Uzsoy, R., 1998. Predictable scheduling of a job shop subject to breakdowns. IEEE Transactions on Robotics and Automation 14, 365–378.
Mehta, S.V., Uzsoy, R., 1999. Predictable scheduling of a single machine subject to breakdowns. International Journal of Computer-Integrated Manufacturing 12, 15–38.
Miyashita, K., 2000. Learning scheduling control knowledge through reinforcements. International Transactions in Operational Research 7, 125–138.
Miyashita, K., Sycara, K., 1995. CABINS: A framework of knowledge acquisition and iterative revision for schedule improvement and reactive repair. Artificial Intelligence 76, 377–426.
Mohan, S., Clancy, D., 1990. SIS—Rule based software for automating job dispatch on the factory floor. In: Proceedings of the Third International Conference on Expert Systems and the Leading Edge in Production Planning and Control, Hilton Head, NC.
Monostori, L., Szelke, E., Kadar, B., 1998. Management of changes and disturbances in manufacturing systems. Annual Reviews in Control 22, 85–97.
Muhlemann, A.P., Lockett, A.G., Farn, C.-K., 1982. Job shop scheduling heuristics and frequency of scheduling. International Journal of Production Research 20, 227–241.
Musselman, K., Uzsoy, R., 2001. Advanced planning and scheduling for manufacturing. In: Salvendy, G. (Ed.), Handbook of Industrial Engineering. John Wiley.
Najmi, A., Lozinski, C., 1989. Managing factory productivity using object-oriented simulation for setting shift production targets in VLSI manufacturing. In: Proceedings of the Autofact Conference, Society of Manufacturing Engineers, pp. 3.1–3.14.
O'Donovan, R., McKay, K.N., Uzsoy, R., 1999. Predictable scheduling on a single machine with machine breakdowns and sensitive jobs. International Journal of Production Research 37, 4217–4233.
O'Kane, J.F., 2000. A knowledge-based system for reactive scheduling decision-making in FMS. Journal of Intelligent Manufacturing 11 (5), 461–474.
Ovacik, I.M., Uzsoy, R., 1997. Decomposition Methods For Complex Factory Scheduling Problems. Kluwer Academic Publishers.
Perry, C.N., Uzsoy, R., 1993. Reactive scheduling of a semiconductor testing facility. In: Proceedings of the International Electronics Manufacturing Technology Symposium, Santa Clara, CA, pp. 191–194.
Pinedo, M., Singer, M., 1999. A shifting bottleneck heuristic for minimizing the total weighted tardiness in a job shop. Naval Research Logistics 46, 1–17.
Piramuthu, S., Park, S.-C., Raman, N., Shaw, M.J., 1991. Integration of simulation modelling and inductive learning in an adaptive decision support system. In: Boczelc, A., Whinston, A. (Eds.), Model Management Systems. IEEE Society Press.
Pritsker, A.A.B., Snyder, K., 1997. Production scheduling using FACTOR. In: Artiba, A., Elmaghraby, S.E. (Eds.), The Planning and Scheduling of Production Systems. Chapman and Hall.
Qi, J.G., Burns, G.R., Harrison, D.K., 2000. The application of parallel multipopulation genetic algorithms to dynamic job shop scheduling. International Journal of Advanced Manufacturing Technology 16, 609–615.
Ramasesh, R., 1990. Dynamic job shop scheduling: A survey of simulation research. OMEGA 18, 43–57.
Rardin, R.L., Uzsoy, R., 2001. Experimental evaluation of heuristic optimization algorithms: A tutorial. Journal of Heuristics 7, 261–304.
Reinfeld, N.V., 1959. Production Control. Prentice-Hall, Englewood Cliffs, NJ.
Roundy, R.O., Maxwell, W.L., Herer, Y.T., Tayur, S.R., Getzler, A.W., 1991. A price directed approach to real-time scheduling of production operations. IIE Transactions 23, 149–160.

Roy, B., Sussmann, B., 1964. Les Problemes d'Ordonnancement avec Contraintes Disjonctives. Note DS No. 9 bis, SEMA, Montrouge.

Sabuncuoglu, I., Bayiz, M., 2000. Analysis of reactive scheduling problems in a job shop environment. European Journal of Operational Research 126, 567–586.

Sabuncuoglu, I., Karabuk, S., 1998. Analysis of the frequency of rescheduling in an FMS with stochastic processing times and machine breakdowns. Department of Industrial Engineering. Bilkent University, Turkey.

Sadeh, N., Otsuka, S., Schnelbach, R., 1993. Predictive and reactive scheduling with the MICRO-BOSS production scheduling and control system. In: Proceedings of the IJCAI Workshop on Knowledge-Based Production Planning, Scheduling and Control, Chambery, France.

Shafaei, R., Brunn, P., 1999a. Workshop scheduling using practical (inaccurate) data part I: The performance of heuristic scheduling rules in a dynamic job shop environment using a rolling time horizon approach. International Journal of Production Research 37, 3913–3925.

Shafaei, R., Brunn, P., 1999b. Workshop scheduling using practical (inaccurate) data, part II: An investigation of the robustness of scheduling rules in a dynamic and stochastic environment. International Journal of Production Research 37, 4105–4117.

Singer, M., 2000. Forecasting policies for scheduling a stochastic due date job shop. International Journal of Production Research 38, 3623–3637.

Singer, M., 2001. Decomposition methods for large job shops. Computers and Operations Research 28, 193–207.

Smith, S.F., 1993. Knowledge-based production management: Approaches, results and prospects. Production Planning and Control 3, 350–380.

Smith, S.F., 1994. OPIS: A methodology and architecture for reactive scheduling. In: Zweben, M., Fox, M.S. (Eds.), Intelligent Scheduling. Morgan Kaufmann, San Fransisco, CA.

Smith, S.F., Keng, N., Kempf, K., 1990a. Exploiting Local Flexibility During Execution of Pre-Computed Schedules. The Robotics Institute, Carnegie Mellon University and Knowledge Based Laboratory, Intel Corporation.

Smith, S.F., Ow, P.S., Muscettola, N., Potvin, J.-Y., Matthys, D.C., 1990b. An integrated framework for generating and revising factory schedules. Journal of the Operational Research Society 41, 539–552.

Sun, D., Lin, L., 1994. A dynamic job shop scheduling framework: A backward approach. International Journal of Production Research 32, 967–985.

Suresh, V., Chaudhuri, D., 1993. Dynamic scheduling—A review. International Journal of Production Economics 32, 53–63.

Szelke, E., Kerr, R.M., 1994. Knowledge-based reactive scheduling. Production Planning and Control 5, 124–145.

Szelke, E., Markus, G., 1995. A blackboard based perspective of reactive scheduling. In: Kerr, R., Szelke, E. (Eds.), Artificial Intelligence in Reactive Scheduling. Chapman and Hall, London, pp. 60–77.

Unal, A.T., Uzsoy, R., Kiran, A.S., 1997. Rescheduling on a single machine with part-type dependent setup times and deadlines. Annals of Operations Research 70, 93–113.

Uzsoy, R., Church, L.K., Ovacik, I.M., Hinchman, J., 1993. Performance evaluation of dispatching rules for semiconductor testing operations. Journal of Electronics Manufacturing 3, 95–105.

Wan, Y.-W., 1995. Which is better, off-line or real-time scheduling. International Journal of Production Research 33, 2053–2059.

Wiers, V.C.S., 1997. Human–Computer Interaction in Production Scheduling: Analysis and Design of Decision Support Systems for Production Scheduling Tasks. Ph.D. Thesis, Eindhoven University of Technology, Ponsen & Looijen, Wageningen, The Netherlands.

Wohlwend, H. et al., 1996. Practices and reliability within the semiconductor industry: A SEMATECH white paper. Technology Transfer #96033106-TR, SEMATECH, Austin, TX.

Wu, H.-H., Li, R.-K., 1995. A new scheduling method for computer based scheduling systems. International Journal of Production Research 33, 2097–2110.

Wu, S.D., Wysk, R.A., 1989. An application of discrete-event simulation to on-line control and scheduling of flexible manufacturing. International Journal of Production Research 27 (9).

Wu, S.D., Byeon, E., Storer, R.H., 1999. A graph-theoretic decomposition of job shop scheduling problems to achieve scheduling robustness. Operations Research 47, 113–124.

Yamamoto, M., Nof, S.Y., 1985. Scheduling/rescheduling in a manufacturing operating system environment. International Journal of Production Research 23, 705–722.

Yellig, E.J., Mackulak, G.T., 1997. Robust deterministic scheduling in stochastic environments: The method of capacity hedge points. International Journal of Production Research 35, 369–379.

Younger, J., 1930. Work Routing in Production Including Scheduling and Dispatching. Ronald, New York.

Zweben, M., Davis, E., Daun, B., Deale, M.J., 1993. Scheduling and rescheduling with iterative repair. IEEE Transactions on Systems, Man and Cybernetics 23, 1588–1596.

Zweben, M., Daun, B., Davis, E., Deale, M., 1994. Scheduling and rescheduling with iterative repair. In: Zweben, M., Fox, M. (Eds.), Intelligent Scheduling. Morgan Kaufman, San Francisco.