Journal of Scheduling **6:** 39–62, 2003.

RESCHEDULING MANUFACTURING SYSTEMS: A FRAMEWORK OF STRATEGIES, POLICIES, AND METHODS

GUILHERME E. VIEIRA[1], JEFFREY W. HERRMANN[2,*], AND EDWARD LIN[3]

[1] *Department of Control and Industrial Automation Engineering, Catholic University of Parana, Curitiba, Parana, Brazil*

[2] *Department of Mechanical Engineering and Institute for Systems Research, University of Maryland, College Park, MD 20742, USA*

[3] *Institute for Systems Research, University of Maryland, College Park, MD 20742, USA*

ABSTRACT

Many manufacturing facilities generate and update production schedules, which are plans that state when certain controllable activities (e.g., processing of jobs by resources) should take place. Production schedules help managers and supervisors coordinate activities to increase productivity and reduce operating costs. Because a manufacturing system is dynamic and unexpected events occur, rescheduling is necessary to update a production schedule when the state of the manufacturing system makes it infeasible. Rescheduling updates an existing production schedule in response to disruptions or other changes. Though many studies discuss rescheduling, there are no standard definitions or classification of the strategies, policies, and methods presented in the rescheduling literature. This paper presents definitions appropriate for most applications of rescheduling manufacturing systems and describes a framework for understanding rescheduling strategies, policies, and methods. This framework is based on a wide variety of experimental and practical approaches that have been described in the rescheduling literature. The paper also discusses studies that show how rescheduling affects the performance of a manufacturing system, and it concludes with a discussion of how understanding rescheduling can bring closer some aspects of scheduling theory and practice.

KEY WORDS: rescheduling; predictive-reactive scheduling; dynamic scheduling

1. INTRODUCTION

Many manufacturing facilities generate and update production schedules, which are plans that state when certain controllable activities (e.g., processing of jobs by resources) should take place. In dynamic, stochastic manufacturing environments, managers, production planners, and supervisors must not only generate high-quality schedules but also react quickly to unexpected events and revise schedules in a cost-effective manner. These events, generally difficult to take into consideration while generating a schedule, disturb the system, generating considerable differences between the predetermined schedule and its actual realization on the shop floor.

*Correspondence to: Dr Jeffrey W. Herrmann, Department of Mechanical Engineering, University of Maryland, College Park, MD 20742. E-mail: jwh2@eng.umd.edu

Contract/grant sponsor: CAPES; Contract/grant number: 3135/95-3

Rescheduling is then practically mandatory in order to minimize the effect of such disturbances in the performance of the system. There are many types of disturbances that can upset the plan, including machine failures, processing time delays, rush orders, quality problems, and unavailable material. As Bean et al. (1991) state, rescheduling is a dynamic approach that responds to disruptions, yet it considers future information (by creating a plan for the future).

In practice, rescheduling is done periodically to plan activities for the next time period based on the state of the system. It is also done occasionally in response to significant disruptions. Because time estimates are incorrect and unexpected events occur, precisely following a schedule becomes more difficult as time passes. In some cases, the system may follow the sequence that the schedule specifies even though the planned start and end times are no longer feasible. Eventually, however, a new schedule will be needed.

A great deal of effort has been spent developing methods to generate optimal production schedules, and countless papers discussing this topic have appeared in scholarly journals. Typically, such papers formulate scheduling as a combinatorial optimization problem.

Many studies of production scheduling problems have employed a standard three-field classification scheme (Graham et al., 1979). This scheme represents a scheduling problem as a triple $\alpha|\beta|\gamma$, where α represents the scheduling environment, β represents any distinctive characteristics of the jobs to be scheduled, and γ describes the objective function. It has been used to describe concisely a wide variety of standard one-machine, parallel machine, and shop scheduling problems, and researchers have also employed it as notation for describing many other static scheduling problems. Many classifications of static scheduling problems have been done (e.g., Herrmann, Lee, and Snowden, 1993), but these do not consider the rescheduling context involved. Liu and MacCarthy (1996) present a classification scheme for scheduling problems in flexible manufacturing systems. This scheme includes a descriptor for aspects of the production management environment, including whether orders are handled periodically or continuously. Nof and Grant (1991) present a framework for real-time control of automated manufacturing systems. Their review covers a variety of schedule generation approaches, including artificial intelligence and knowledge-based approaches. For a single machine operating in a dynamic, stochastic environment, Markowitz and Wein (2001) classify scheduling problems based on three attributes: the presence of setups, the presence of due dates, and the type of products (standardized or customized).

However, the scope of papers on rescheduling varies greatly, and there is no standard classification scheme. In the literature on rescheduling, there are three primary types of studies: one, methods for repairing a schedule that has been disrupted; two, methods for creating a schedule that is robust with respect to disruptions; and three, studies of how rescheduling policies affect the performance of the dynamic manufacturing system. To understand this work, this paper presents a framework for understanding rescheduling not only as a collection of techniques for generating and updating production schedules but also as a control strategy that has an impact on manufacturing system performance in a variety of environments. The framework includes rescheduling environments, rescheduling strategies, rescheduling policies, and rescheduling methods. The rescheduling environment identifies the set of jobs that need to be scheduled. A rescheduling strategy describes whether or not production schedules are generated. A rescheduling policy specifies when rescheduling should occur. Rescheduling methods describe how schedules are generated and updated.

This paper defines these concepts and reviews papers that describe specific approaches in each area. Because of the lack of standardized definitions and classifications, this paper will also

define common scheduling terms. However, this paper is not a comprehensive survey of the numerous approaches used to generate or repair schedules.

Finally, the paper discusses studies that show how rescheduling affects the performance of a manufacturing system, and it concludes with a discussion of how understanding rescheduling can bring closer some aspects of scheduling theory and practice.

The remainder of the paper is organized as follows. The next section describes rescheduling in general, along with the types of events that cause rescheduling. Section 3 defines scheduling terms, presents the rescheduling framework, and discusses rescheduling environments. Section 4 discusses performance measures. Section 5 describes rescheduling strategies and rescheduling policies. Section 6 explains methods for generating robust schedules and methods for updating schedules. Section 7 describes how rescheduling policies affect manufacturing system performance. Section 8 discusses the gap between scheduling theory and practice. Section 9 concludes the paper and presents some lessons learned from this effort.

2. RESCHEDULING MANUFACTURING SYSTEMS

Manufacturing facilities are complex, dynamic, stochastic systems. From the beginning of organized manufacturing, workers, supervisors, engineers, and managers have developed many clever and practical methods for controlling production activities. Although dispatching rules, kanban cards, and other decentralized production control policies are in use (Panwalkar and Iskander, 1977; Green and Appel, 1981; Hopp and Spearman, 1996), many manufacturing facilities generate and update production schedules. Such policies are usually quick but myopic because they do not use global information typically.

In manufacturing systems with a wide variety of products, processes, and production levels, production schedules can enable better coordination to increase productivity and minimize operating costs. A production schedule can identify resource conflicts, control the release of jobs to the shop, and ensure that required raw materials are ordered in time. A production schedule can determine whether delivery promises can be met and identify time periods available for preventive maintenance. A production schedule gives shop floor personnel an explicit statement of what should be done so that supervisors and managers can measure their performance.

Note that, after a schedule is generated, manufacturing operations begin. Managers and supervisors want the shop floor to follow the schedule. In practice, operators may deviate from the schedule. Ideally, the schedule is followed as closely as possible. Small deviations from scheduled start times and end times are expected and usually ignored. (The definition of small depends on the facility in question.) Larger deviations or changes to the sequence occur when unexpected events disrupt the initial schedule. Even if the managers and supervisors do not explicitly update the schedule, schedule repair occurs as the operators react to the disruptions, delaying tasks or performing tasks out of order.

Rescheduling is the process of updating an existing production schedule in response to disruptions or other changes. This includes the arrival of new jobs, machine failures, and machine repairs. Rescheduling studies have considered many types of manufacturing systems, including single machine systems (Vieira, Hermann, and Lin, 2000a; Huang, Kanal, and Tripathi, 1990; Ovacik and Uzsoy, 1994), parallel machine systems (Ovacik and Uzsoy, 1995; Vieira, Hermann, and Lin, 2000b), flow shops (McPherson and White, 1998), job shops (Muhlemann, Lockett, and Farn, 1982; Sun and Lin, 1994), and flexible manufacturing cells and systems (Yamamoto and Nof, 1985; Tabe and Salvendy, 1988; Dutta, 1990; Dhingra, Musser

and Blankenship, 1992; Jain and Elmaraghy, 1997; Peng and Chen, 1998; Sabuncuoglu and Karabuk, 1999). Many papers have addressed flexible manufacturing systems because they require tight synchronization between the shop floor and the planned procedures in order to reach the efficiency they are expected to have. Basnet and Mize (1994) review approaches for scheduling (but not rescheduling) and control of flexible manufacturing systems.

Unexpected events (disruptions) can change the system status and affect performance. If it will cause significant deterioration in performance, the event will trigger rescheduling to reduce the impact. For this reason, these events are called *rescheduling factors* (Dutta, 1990; Dhingra, Musser and Blankenship, 1992). The following are the most common factors identified in rescheduling studies:

- Machine failure (Yamamoto and Nof, 1985; Church and Uzsoy, 1992; Li, Shyu, and Adiga, 1993; Kim and Kim, 1994; Abumaizar and Svestka, 1997; Fang and Xi, 1997; Jain and Elmaraghy, 1997; Sabuncuoglu and Karabuk, 1999; Shafaei and Brunn, 1999a, b)
- Urgent (rush or 'hot') job arrival (Li, Shyu, and Adiga, 1993; Kim and Kim, 1994; Abumaizar and Svestka, 1997; Jain and Elmaraghy, 1997)
- Job cancellation [Li, Shyu, and Adiga, 1993; Abumaizar and Svestka, 1997; Jain and Elmaraghy, 1997)
- Due date change (delay or advance) (Li, Shyu, and Adiga, 1993; Henning and Cerda, 1995; Fang and Xi, 1997)
- Delay in the arrival or shortage of materials (Li, Shyu, and Adiga, 1993; Henning and Cerda, 1995; Fang and Xi, 1997)
- Change in job priority (Henning and Cerda, 1995; Jain and Elmaraghy, 1997)
- Rework or quality problems (Church and Uzsoy, 1992; Li, Shyu, and Adiga, 1993)
- Over- or underestimation of process time (Li, Shyu, and Adiga, 1993)
- Operator absenteeism (Church and Uzsoy, 1992)

The above events may trigger other actions (listed below) that, in turn, suggest rescheduling (Li, Shyu, and Adiga, 1993; Henning and Cerda, 1995):

- Overtime
- In-process subcontracting
- Process change or re-routing
- Machine substitution
- Limited manpower
- Setup times
- Equipment release

3. TERMINOLOGY AND FRAMEWORK

In the literature on rescheduling, the inconsistent use of many terms makes understanding the field difficult. This section defines some common terms and presents a rescheduling framework that helps define these terms and show their relationship.

3.1. Definitions

A *manufacturing system* organizes equipment, people, and information to fabricate and assemble finished goods that are shipped to a customer. This system may be as large as a factory

or as small as a manufacturing cell. According to Black (2000), a manufacturing system is "the collection of operations and processes used to produce a desired product." A manufacturing system does not include finance, design engineering, research and development, production and inventory planning, purchasing, or distribution (although the last three items belong in more general manufacturing systems). Note that it does include order release, shop floor control, and material handling.

Order release controls a manufacturing system's input by determining which orders (jobs) should be moved into production. It may be known as job release, order review/release, input/output control, or just input control.

Shop floor control determines which operation each person and piece of equipment should do and when they should do it. In general, this process controls all production and material handling resources. Design decisions include order release policies (including WIP levels for pull systems), dispatching rules, batch sizes, and preventive maintenance policies.

A *production schedule* specifies, for each resource required for production, the planned start time and end time of each job assigned to that resource.

Scheduling is the process of creating a production schedule for a given set of jobs and resources.

Rescheduling is the process of updating an existing production schedule in response to disruptions or other changes. This includes the arrival of new jobs, machine failures, and machine repairs.

The *rescheduling environment* identifies the set of jobs that the schedule should include. A *rescheduling strategy* describes whether or not production schedules are generated. A *rescheduling policy* specifies when and how rescheduling is done. The policy specifies the events that trigger rescheduling. These events may be predictable (even regular) or unpredictable. The policy specifies the method used to revise the existing schedule. Note that the policy may specify different methods for different situations. If these policies have any parameters (for instance, the length of the rescheduling period), the policy specifies these parameters. *Rescheduling methods* generate and update production schedules.

3.2. A rescheduling framework

The scope of rescheduling research varies widely. Some studies discuss comprehensive strategies for rescheduling manufacturing systems, while others present specific techniques for resolving constraint violations that occur when an unexpected event happens. This literature is thus unlike the literature that discusses static scheduling problems. In the latter, the focus of each paper is generally on one (or more) well-defined scheduling problem(s), and each paper establishes the problem's computational complexity, identifies properties of optimal schedules, proves the optimality of an exact solution approach, or compares the performance of heuristic solution approaches experimentally.

Figure 1 presents a framework for understanding rescheduling research. The framework includes rescheduling environments, rescheduling strategies, rescheduling policies, and rescheduling methods. Either rescheduling strategy (dynamic scheduling or predictive-reactive scheduling) can be used in any rescheduling environment with uncertainty or variability. However, dynamic rescheduling environments are the ones most relevant to manufacturing systems, and the predictive-reactive rescheduling strategy is the approach most commonly used in practice. Still, there are a great variety of rescheduling policies used in predictive-reactive

Rescheduling environments				
Static (finite set of jobs)		Dynamic (infinite set of jobs)		
Deterministic (all information given)	Stochastic (some information uncertain)	No arrival variability (cyclic production)	Arrival variability (flow shop)	Process flow variability (job shop)

Rescheduling strategies				
Dynamic (no schedule)		Predictive-reactive (generate and update)		
Dispatching rules	Control-theoretic	Rescheduling policies		
		Periodic	Event-driven	Hybrid

Rescheduling methods				
Schedule generation		Schedule repair		
Nominal schedules	Robust schedules	Right-shift rescheduling	Partial rescheduling	Complete regeneration

Figure 1. Rescheduling framework

scheduling. Section 3.3 discusses rescheduling environments. Section 5 describes the two rescheduling strategies and the rescheduling policies. Rescheduling methods (described in Section 6) are procedures for generating and repairing schedules, which are necessary only for rescheduling policies used in predictive-reactive scheduling. (Dynamic scheduling does not create or update schedules.)

This paper will use this framework to explain the concepts of rescheduling. Although this paper is not a comprehensive survey, this framework can also be used to understand the contributions of individual papers. For example, Table I presents a short list of papers and, for each paper, describes the type of rescheduling approach it studies.

3.3. Rescheduling environments

The rescheduling environment, which identifies the set of jobs that need to be scheduled, is an important component of the rescheduling framework, as Figure 1 illustrates. Static rescheduling environments have a finite set of jobs (Pinedo, 1995; Pinedo and Chao, 1999). Dynamic rescheduling environments have an infinite set of jobs (i.e., jobs continue to arrive over an infinite time horizon) (Church and Uzsoy, 1992; Fang and Xi, 1997).

Deterministic, static scheduling problems can be viewed as a special case of rescheduling, where there is a finite set of jobs and no uncertainty about the future. The specified schedule can be followed without any modifications. (In some cases, these problems may be decomposed into

Table I. Descriptions of selected papers using the rescheduling framework

Reference	Rescheduling environment	Rescheduling strategy	Rescheduling policy	Rescheduling methods	
				Schedule generation	Schedule repair
Vieira, Herrmann, and Lin (2000a)	Dynamic flow shop	Predictive-reactive	Periodic, event-driven	Nominal	Right-shift
Vieira, Herrmann, and Lin (2000b)	Dynamic flow shop	Predictive-reactive	Periodic, hybrid, event-driven	Nominal	Complete regeneration
Farn and Muhlemann (1979)	Dynamic flow shop	Predictive-reactive	Periodic	Nominal	Complete regeneration
Muhlemann, Lockett, and Farn (1982)	Dynamic job shop	Predictive-reactive	Periodic	Nominal	Complete regeneration
Mehta and Uzsoy (1998)	Dynamic job shop	Predictive-reactive	Periodic	Robust	Right-shift
Byeon, Wu, and Storer (1998)	Static, stochastic	Predictive-reactive	Event-driven	Robust	Partial rescheduling
Leon, Wu, and Storer (1994)	Static, stochastic	Predictive-reactive	Event-driven	Robust	Right-shift
Church and Uzsoy (1992)	Dynamic flow shop	Predictive-reactive	Hybrid	Nominal	Complete regeneration
Bierwirth and Mattfeld (1999)	Dynamic job shop	Predictive-reactive	Event-driven	Nominal	Complete regeneration
Wu and Li (1995)	Dynamic job shop	Predictive-reactive	Event-driven	Nominal	Partial rescheduling
Herrmann, Lee, and Hinchman (1995)	Dynamic job shop	Predictive-reactive	Periodic	Nominal	Complete regeneration
Kumar (1994)	Dynamic job shop	Dynamic			
Akturk and Gorgulu (1999)	Static, stochastic	Predictive-reactive	Event-driven	Nominal	Partial rescheduling

subproblems as part of a solution technique, but this does not change the nature of the problem.)

Stochastic, static rescheduling environments are an important special case of rescheduling. Again, there is a finite set of jobs, but some variables are uncertain. For instance, when task processing times are modeled as random variables, a solution may specify resource assignments and task sequences, but the actual task start times and completion times will not match the expected ones. At the minimum, executing the schedule requires some rule or policy for reconciling the error in the schedule. However, other policies exist. One can modify the schedule at some point during execution to react to additional information, or one can construct a solution that only partially specifies the schedule, leaving details unspecified until the appropriate time comes (Wu, Byeon, and Storer, 1999). There also exist problems in which the uncertainty is not modeled as a probability distribution. In this case, worst-case performance is a key objective (e.g., see Daniels and Kouvelis (1995); Herrmann (1999)).

A dynamic rescheduling environment has an infinite stream of jobs. Each job requires scheduling before it can be processed. Three important cases exist.

First, if there is no uncertainty or variability in the arrival process, then the jobs to be processed are known in advance, and the production schedule is continuously repeated. If the jobs can be grouped into a minimal part set that is continuously repeated, then a single scheduling decision is needed to create a sequence of operations that will be continuously repeated. This yields a cyclic scheduling problem. See, for example, Matsuo (1990), Roundy (1992), Kamoun and Sriskandarajah (1993), Hall and Sriskandarajah (1996), and Lee and Posner (1997). Pinedo and Chao (1999) describe the use of cyclic scheduling in flexible assembly systems. If the lot sizes are not specified, then the production schedule is a cycle of production runs that must be determined by solving the economic lot scheduling problem (e.g., see Pinedo and Chao (1999)).

Second, there may exist some uncertainty in job arrivals, but all jobs follow the same route through the manufacturing system, and the arrival rate is steady. When there exist significant setups between different classes of jobs or reentrant flow, scheduling is necessary to determine when a resource should switch from processing one type of job to another (Mehta and Uzsoy, 1998; Markowitz and Wein, 2001).

Third, there may exist process flow variability along with the variability in job arrivals. Job shops often have this characteristic, since there are many products, but a limited subset of them are being produced at any given time. Thus, a specific product's arrival process has great variability. In some situations, no advance information is available about jobs before they arrive. Otherwise, some information about future arrivals may be known, but the information is subject to change as new jobs are added and existing jobs are delayed or deleted (Church and Uzsoy, 1992; Abumaizar and Svestka, 1997).

Another aspect that characterizes rescheduling environments is the presence of potential additional capacity using subcontracting or overtime (see, for instance, Akkan (1996), Arslan, Ayhan, and Olsen (2001). A facility that works 24 h every day is not the same as a facility that work five 8-h shifts a week. The presence of capacity buffers (in the form of potential overtime) affects the rescheduling environment, since it relaxes one set of constraints (capacity) but adds another set of costs (overtime).

3.4. Other terms

This section defines some other terms used to describe aspects of rescheduling policies: scheduling point (or rescheduling point), rescheduling period, rescheduling frequency, scheduling stability, scheduling robustness, and scheduling nervousness.

A *scheduling point* (or *rescheduling point*) is the point in time when a scheduling decision is made (Sabuncuoglu and Karabuk, 1999). That is, it is the point in time when a schedule is created or revised.

The *rescheduling period* is the time between two consecutive scheduling points. The *rescheduling frequency*, therefore, is the inverse of the rescheduling period and it measures how often rescheduling is performed.

Scheduling stability measures the number of revisions or changes that a schedule undergoes during execution (Church and Uzsoy, 1992; Wu, Storer, and Chang, 1993). Cowling and Johansson (2002) describe a general approach for measuring schedule stability and present a specific stability measure that calculates the average absolute change of the start and completion

times of two schedules. *Scheduling nervousness* was originally mentioned in the context of material requirement planning (MRP) systems, where it was defined as "significant changes in MRP plans" or "instability" (Vollmann, Berry, and Whybark, 1997). Because nervousness is constant change in the schedule (frequent rescheduling), it is the opposite of schedule stability. A "nervous" system presents little predictability. A rescheduling policy that yields fewer revisions increases schedule stability (and decreases schedule nervousness).

Schedule robustness measures how much disruptions would degrade the performance of the system as it executes the schedule. Stability and nervousness measure the changes to a schedule, but robustness measures the changes to system-level performance.

4. PERFORMANCE MEASURES

A variety of performance measures guide rescheduling. These measures can be separated into three groups (Wu, Storer, and Chang, 1993; Jain and Elmaraghy, 1997; Shafaei and Brunn, 1999a): measures of schedule efficiency, measures of schedule stability, and cost.

Measures of schedule efficiency are often used when generating a production schedule. They are generally time-based measures (Shafaei and Brunn, 1999a): makespan (Yamamoto and Nof, 1985; Wu, Storer, and Chang, 1993; Fang and Xi, 1997; Vollmann, Berry, and Whybark, 1997; Mehta and Uzsoy, 1998; Sabuncuoglu and Karabuk, 1999), mean tardiness (Tabe and Salvendy, 1988; Kim and Kim, 1994; Henning and Cerda, 1995; Jain and Elmaraghy, 1997; Sabuncuoglu and Karabuk, 1999), mean flow-time (Kim and Kim, 1994; Jain and Elmaraghy, 1997; Vieira, Herrmann, and Lin, 2000a, b), average resource utilization (Tabe and Salvendy, 1988; Henning and Cerda, 1995; Jain and Elmaraghy, 1997), and maximum lateness (Church and Uzsoy, 1992).

Schedule stability is not an issue in static, deterministic rescheduling environments since the schedule does not need updating. However, in other rescheduling environments, stability, nervousness, and robustness are important measures. Wu, Storer, and Chang, 1993, for instance, have said that the impact of schedule change is a nonregular performance measure defined in two ways: (1) the starting time deviations between the new schedule and the original schedule, and (2) a measure of the sequence difference between the two schedules. Abumaizar and Svestka (1997) proposed similar ideas saying that measures of stability deal with deviation from the initial schedule. Watatani and Fujii (1992) and Dhingra, Musser, and Blankenship (1992) also considered the deviation between the revised and initial schedules as performance measures, even though they did not call it schedule stability.

The impact of machine failure seems to be the major concern when searching for more stable (less nervous) and robust schedules. Shafaei and Brunn (1999b) have addressed the robustness of scheduling rules in a dynamic and stochastic environment. They concluded that as the level of uncertainty increases, frequent rescheduling becomes more effective in improving the robustness of the schedules. Wu, Storer, and Chang (1993) have studied rescheduling heuristics using schedule efficiency (makespan) and schedule stability as performance measure criteria. For the single-machine system they have considered, the heuristic used generated stable schedules while retaining near-optimal makespans.

Time-based performance measures (measures to reach schedule efficiency) do not completely reflect the economic performance of the manufacturing system. So, due to the lack of an overall, efficient, time-based performance measure, researchers have recognized that the scheduling decisions should also be evaluated using an economic performance measure. The objective then is to minimize the cost of starting jobs too early, work-in-process inventory, and tardiness.

Issues such as job profitability, total cost minimization, reduction in WIP, and the cost of missed due dates are more important for managers than the time-based measures mentioned above (Shafaei and Brunn, 1999a, b). Shafaei and Brunn (1999a, b) have proposed the use of a total cost function in terms of job due date, completion time, number of jobs, number of operations, operation processing time, job raw material cost, processing cost of operations, job revenue, processing start times, job release time, job tardiness, holding cost rate, and tardiness cost rate.

In general, rescheduling costs occur in three categories: computational costs, setup costs, and transportation costs. Computational costs include the computational burden on the computer running the scheduling system (Church and Uzsoy, 1992; Sabuncuoglu and Karabuk, 1999), the nonrecurring costs of investments in the necessary information systems (sensors, displays, communication networks, hardware, and software), and the recurring costs of administration, maintenance, and upgrades. If rescheduling is done manually, then the computational cost includes the time that the planners, managers, and supervisors spend generating and updating schedules. Setup costs occur when tooling and fixtures are created or allocated in advance according to the schedule. Thus, a change in the schedule will incur costs to reallocate pallets and replan the tools (Olumolade and Norrie, 1996). Transportation costs (also called material handling costs) are related to delivering materials earlier than required or additional material handling work to transport jobs from one scheduled machine to other points in the shop (Olumolade and Norrie, 1996). For instance, Bean et al. (1991) use the number of jobs reassigned as a measure of solution cost that must be balanced against tardiness costs and computational effort.

In dynamic rescheduling environments, the relative values of the rescheduling period and the mean total processing time requirements of a job will affect the performance measures used in predictive-reactive rescheduling. When the rescheduling period is relatively large, jobs can be started and completed between rescheduling events. Scheduling objectives will typically focus on completing the available jobs within that time period. When the rescheduling period is relatively small, the system will have, at each rescheduling point, some jobs that are available and waiting to start and many others that started during a previous period but still require more processing. In a job shop environment, scheduling objectives are much more complex, since there is a need to balance available capacity among jobs at different stages in their processing. This is especially true in shops with re-entrant flow, like those found in semiconductor wafer fabrication plants (Kempf, 1994; Kumar, 1994).

5. RESCHEDULING STRATEGIES

This section describes two common strategies for controlling production in dynamic rescheduling environments that have uncertain job arrivals. The two strategies are dynamic scheduling and predictive-reactive scheduling. Predictive-reactive scheduling includes three types of rescheduling policies: periodic, event-driven, and hybrid.

5.1. Dynamic scheduling

Dynamic scheduling does not create production schedules. Instead, decentralized production control methods dispatch jobs when necessary and use information available at the moment of dispatching. Such schemes use dispatching rules or other heuristics to prioritize jobs waiting for

processing at a resource (Perkins and Kumar, 1989; Church and Uzsoy, 1992; Fang and Xi, 1997). Some authors refer to dynamic scheduling schemes as online scheduling or reactive scheduling (Li, Shyu, and Adiga, 1993; Olumolade and Norrie, 1996; Sabuncuoglu and Karabuk, 1999).

Dispatching rules and pull mechanisms are used to control production without a production schedule. When a machine becomes available, it chooses from among the jobs in its queue by using a dispatching rule that sorts the jobs by some criteria. Common dispatching rules employ processing times and due dates in simple rules and complex combinations. Some dispatching rules are extensions of policies that work well on simple machine scheduling problems (e.g., Shortest Processing Time (SPT) and Earliest Due Date (EDD)). The computational effort of dispatching rules is low when simple rules (like SPT or EDD) are used. However, some dispatching rules require a large amount of information, and the job priorities must be recalculated at every dispatching decision.

Panwalkar and Iskander (1977) provide an extensive list of dispatching rules. They categorize these rules into five classes: simple dispatching rules, combinations of simple rules, weighted priority indexes, heuristic scheduling rules, and other rules.

Green and Appel (1981) examine the problem of job shop scheduling by asking the following questions: What traditional dispatching rules do experienced schedulers select? Would dispatch rule selection be influenced by urgency? Would schedulers select a dispatch order based on organizational influence or peer pressure? The authors asked schedulers in a number of plants to denote which of the following rules they used: due date, slack, operation due date, slack per operation, SPT, FCFS, COVERT, Program in Greatest Trouble (PGT), or friend needs a favor (FNF). The authors report that influence systems affect scheduling. The PGT rule (a coalition rule) was highly valued, but FNF (an individual rule) was rejected. Traditional and theoretical rules were not highly valued.

Pull mechanisms such as kanban cards and constant WIP (CONWIP) order release policies add production authorization cards to the system so that a resource can work only when both material and cards are available. Hopp and Spearman (1996) provide a good introduction to these topics. Buzacott and Shanthikumar (1993) analyze a generalized production authorization policy.

Dynamic scheduling is closely related to real-time control, since decisions are made based on the current state of the manufacturing system. Controlling a manufacturing system so that it maintains a desired inventory position (in work-in-process or finished goods) is a common strategy when there is steady demand for each product. There may be multiple process flows (routes), but they are known, and each one has a steady throughput of jobs following that flow. This consistency makes base stock policies, hedging points, kanban, and other pull-based mechanisms feasible. The system works to maintain a low level of work-in-process, but the consistent demand ensures that this inventory turns over regularly. See, for example, Hopp and Spearman (1996), Gershwin (1994), and Bispo and Tayur (2001).

Gershwin (1994) reviews literature of control theoretic models of manufacturing systems. The models are used to develop rules for deciding which action to take and when to take it in response to random disruptions. For instance, these control policies can be implemented as dispatching rules or hedging-point policies.

In a number of papers, Kumar and others (e.g., Kumar (1994); Perkins and Kumar (1989); Chase and Ramadge (1992)) have studied the control of dynamic manufacturing systems. Specifically, they have described classes of dispatching rules that identify which waiting job a

resource should process next. For machines without setup times, the proposed dispatching rules are a class of least slack policies that prioritize each job by the difference between its due date (or some surrogate) and the expected amount of time until the job is completed. For resources with setup times, the proposed dispatching rules focus on completing all waiting jobs of one type before performing a setup and processing jobs of another type. All of the rules studied keep a machine working if there are any jobs waiting for processing. (That is, the machine cannot ignore waiting jobs.) Kumar (1994) summarizes the results of work on the stability and performance of these policies. This important work demonstrates why certain classes of dispatching rules work well and provides guidance when selecting dispatching rules. However, there exist dynamic manufacturing systems for which these types of dispatching rules are inappropriate or suboptimal. For example, Chase and Ramadge (1992) demonstrated that there exist idling policies that have superior performance. For a single machine operating in a dynamic, stochastic environment, Markowitz and Wein (2001) present dynamic cyclic policies that minimize the long-run expected average costs of earliness, tardiness, holding, and setups.

5.2. Predictive-reactive scheduling

Predictive-reactive scheduling is a common strategy to rescheduling dynamic manufacturing systems (Huang, Kanal, and Tripathi, 1990; Dhingra, Musser, and Blankenship, 1992; Szelke and Kerr, 1994; Henning and Cerda, 1995; Jain and Elmaraghy, 1997; Jones, Riddick, and Rabelo, 1998; Mehta and Uzsoy, 1998). Predictive-reactive scheduling has two primary steps. The first step generates a production schedule. The second step updates the schedule in response to a disruption or other event to minimize its impact on system performance (Yamamoto and Nof, 1985; Church and Uzsoy, 1992; Abumaizar and Svestka, 1997; Sabuncuoglu and Karabuk, 1999). Rescheduling can occur frequently in a dynamic rescheduling environment, or it can simply be a single revision of the schedule of a stochastic, static rescheduling environment.

Some studies have described approaches for generating robust schedules that will perform well even if disruptions occur. Section 6.1 will describe these approaches in more detail. Section 6.2 describes approaches used to update (repair) a schedule when a disruption occurs.

Predictive-reactive scheduling is an iterative process. Wu and Li (1995) have described rescheduling as an iterative process of three steps: The evaluation step evaluates the impact that a disruption causes. No further action is required if the impact is acceptably small. (This might be the case if there is sufficient idle time in the system to absorb the negative impact of the disruption (Sabuncuoglu and Karabuk, 1999).) The solution step determines the rescheduling solutions that can enhance the performance of the existing schedule. However, determining the best rescheduling solution still remains an open research issue and, consequently, is the most difficult part of the rescheduling process. The revision step updates the existing production schedule or generates a new one. If the result is unacceptable, the solution step must be revisited.

Yamamoto and Nof (1985) have proposed a rescheduling approach following a general three-phase scheme. The planning phase constructs an initial schedule just prior to the start of a new work period, based on all available production requirements. It prepares the information necessary for the operations during a given period. The control phase compares the actual progress of operations to the current schedule every time a new operation begins or finishes. If the difference exceeds a specified limit, the rescheduling phase should begin. The rescheduling phase constructs a revised schedule considering the operational changes that have triggered the

rescheduling. For instance, if a new part mix is required, then a revised schedule is required. When a machine fails, the expected duration of the breakdown has to be considered. The scheduling procedure itself is essentially the same as in the planning phase.

A rescheduling policy is needed to implement a predictive-reactive scheduling strategy. Three types of rescheduling policies have been studied: periodic, event-driven, and hybrid. The periodic and hybrid strategies have received special attention under the name of *rolling time horizon* approaches (Church and Uzsoy, 1992; Fang and Xi, 1997). When scheduling is performed on a rolling time horizon, the overall scheduling problem is decomposed into smaller and static scheduling problems.

A periodic policy reschedules the facility periodically and implements the schedules on a rolling time horizon basis (Baker and Peterson, 1979, Muhlemann, Lockett, and Farn, 1982; Church and Uzsoy, 1992; Ovacik and Uzsoy, 1994, 1995; Fang and Xi, 1997; McPherson and White, 1998; Sabuncuoglu and Karabuk, 1999; Vieira, Herrmann, and Lin, 2000a, b). Church and Uzsoy (1992) provide a good detailed explanation of this rescheduling policy. In many industrial situations, scheduling is done on a periodic basis, especially in environments where there is no online data acquisition from the shop floor to monitor the state of the plant in real time. In these environments, a scheduler will gather all available information from the shop floor and higher level control systems in order to develop schedules at regular intervals. The schedule will then be implemented and not revised until the next period begins. This periodic approach yields more schedule stability and less schedule nervousness than constant rescheduling. Unfortunately, following an established schedule in the face of significant changes in the system status may compromise performance. Determining the optimal rescheduling period is also a difficult task when using this type of policy.

Prietula et al. (1994) describe a rolling horizon scheduling system that creates a five-week schedule every week. A human expert and the scheduling software (which uses a knowledge base and a search algorithm) collaborate to generate a schedule. Kempf (1994) describes a predictive-reactive scheduling approach for semiconductor wafer fabrications and an artificial intelligence (AI)-based tool that generates a production schedule each shift. Herrmann, Chung-Yee Lee, and Hinchman (1995) describe a periodic policy that uses a genetic algorithm to find good job shop schedules at the beginning of each shift in a semiconductor test facility. The genetic algorithm finds solutions that minimize the number of tardy jobs.

Short-interval scheduling seeks to monitor production multiple times each shift. A production schedule specifies what each workstation will do during the shift. After a few hours, the supervisor will check the schedule to determine if the workstation is meeting the schedule. If not, the supervisor and the operators can decide on corrective actions. This repeats at intervals less than a shift. For a more complete description see, for instance, Koenig (2000).

In an event-driven rescheduling policy, rescheduling can happen repeatedly in dynamic manufacturing systems or it can simply be a single event to revise a schedule in a static system. Most of the work in rescheduling uses this strategy in a static environment, generally to reschedule the system when machine failures occur (Yamamoto and Nof, 1985; Church and Uzsoy, 1992; Li, Shyu, and Adiga, 1993; Kim and Kim, 1994; Abumaizar and Svestka, 1997; Fang and Xi, 1997; Jain and Elmaraghy, 1997; Sabuncuoglu and Karabuk, 1999; Shafaei and Brunn, 1999a, b). For dynamic rescheduling environments, Vieira, Herrmann, and Lin, (2000a, b) studied event-driven rescheduling policies that trigger rescheduling when the total number of job arrivals reaches a threshold. Bierwirth and Mattfeld (1999) study a rescheduling policy that creates a new schedule every time a new job arrives. In OPIS (Smith et al., 1990;

Smith, 1994), the rescheduling triggers include time conflicts, capacity conflicts, and rescheduling opportunities that occur when external events create additional capacity.

In the extreme, a new schedule is created (or revised) every time an event that alters system status occurs (Church and Uzsoy, 1992). Clearly the time spent doing rescheduling can become excessive and, more than the other strategies, it will require a fast and reliable electronic data collection to quickly capture new events. Unfortunately, in large facilities, with many events occurring in rapid succession, the system may be in a permanent state of rescheduling, with high nervousness (low stability) and excessive computational requirements.

A hybrid rescheduling policy (Church and Uzsoy, 1992; Kim and Kim, 1994; Fang and Xi, 1997; Vieira, Herrmann, and Lin, 2000a, b) reschedules the system periodically and also when special (or major) events take place. Major events are usually machine breakdowns, but they can also be the arrival of urgent jobs, job cancellation, or job priority changes. Chacon (1998) described a system being used at Sony Semiconductor that uses periodic scheduling with manual rescheduling in case an unscheduled event makes the schedule significantly obsolete. Church and Uzsoy (1992) discussed a hybrid policy that revises the schedule at the beginning of each time period and when significant disruptions occur. Vieira, Herrmann, and Lin, (2000b) studied a hybrid rescheduling policy that triggers rescheduling when a machine fails and when a repair is completed.

6. RESCHEDULING METHODS

This section describes methods used, as part of predictive-reactive scheduling, to create or update schedules. Schedule generation methods include most of the literature in the area of scheduling and are beyond the scope of this paper. Interested readers should see Pinedo and Chao (1999), Pinedo (1995), or similar introductory texts on production scheduling. This section will concentrate on methods that generate robust schedules and methods that update schedules in response to a disruption, since these approaches are most closely related to rescheduling. Browne (1989), Zweben and Fox (1994), and Brown and Scherer (1995) include works describing a wide range of knowledge-based and AI approaches for generating and updating production schedules.

Also related are papers that describe approaches for static, stochastic rescheduling environments. Typically, these approaches are used to generate an initial schedule that optimizes the expected performance. The execution of such a schedule will require some technique for repair. For an overview of stochastic scheduling problems, see, for example, Pinedo (1995).

6.1. Generating robust schedules

Rescheduling is a necessary reaction to disruptions. Simple schedule adjustments (like right shifts, discussed in Section 6.2) require little effort and are easy to implement. However, they may lead to poor system performance compared to more extensive schedule changes. Generating robust schedules is an attempt to maintain good system performance with simple schedule adjustments.

A number of papers have proposed methods for creating schedules that are robust with respect to disruptions. Leon, Wu, and Storer (1994) analyze how a single disruption delays a job shop schedule and present surrogate measures for estimating that delay in more general cases.

They present a genetic algorithm to find robust schedules that minimize expected delay and expected makespan. Byeon, Wu, and Storer (1998) and Wu, Byeon, and Storer (1999) present approaches to create robust partial schedules for a job shop that is subject to disturbances. Byeon, Wu, and Storer (1998) decompose the job shop scheduling problem and solve a variant of the generalized assignment problem. Wu, Byeon, and Storer (1999) use a branch-and-bound algorithm to process the corresponding disjunctive graph and form a partial schedule. The incomplete portions of the schedule are resolved at the appropriate time, giving the shop some flexibility to handle disruptions. Their results show that, in a range of situations, such a schedule leads to better system performance than dispatching rules. However, as the amount of processing time variability increases, dispatching rules led to better performance. Similarly, Mehta and Uzsoy (1998) present an approach to create predictive schedules that include inserted idle time as a means to reduce the impact of disruptions. The method uses the shifting bottleneck algorithm to form operation sequences and then inserts idle time using a construction heuristic. Their studies indicated that schedules that are robust to stochastic disturbances could be generated without much degradation of system performance. Their results however, did not consider the effects of finishing jobs too early (when breakdowns do not occur).

When probability distributions are not available or appropriate (e.g., for a risk-averse decision-maker), worst-case performance is a key objective. Daniels and Kouvelis (1995) and Herrmann (1999) develop approaches for optimizing worst-case performance of production schedules. Daniels and Kouvelis (1995) use an enumeration technique, while Herrmann (1999) presents a genetic algorithm.

O'Donovan, Uzsoy, and McKay (1999) describe methods, based on careful observation of scheduling practice, that generate schedules that are robust with respect to machine breakdowns. The scheduling objective is to minimize the expected deviation in completion times (the difference between the planned completion times and the realized completion times) as well as to minimize expected tardiness on a one-machine scheduling problem with nonzero release dates. The approach, similar to Mehta and Uzsoy (1998), first uses a dispatching rule to generate a schedule and then a simple policy to insert idle time between jobs based on expected downtime. The paper also describes a slightly different version of the method that, while generating a robust schedule, considers the impaired condition that the repaired machine will have after any failures. Experimental results show that these approaches improve schedule robustness with little impact on other performance measures.

Shafaei and Brunn (1999b) investigated the robustness of a number of scheduling rules in a dynamic, stochastic job shop. Schedules were created periodically using a non-delay scheduling algorithm and one of seven scheduling rules. Their results indicated that as the level of uncertainty increases, frequent rescheduling becomes more effective in improving the robustness of the schedule.

6.2. Repairing schedules

After a schedule is generated, manufacturing operations begin. Managers and supervisors want the shop floor to follow the schedule. In practice, operators may deviate from the schedule. Ideally, the schedule is followed as closely as possible. Small deviations from scheduled start times and end times are expected and usually ignored. (The definition of small depends on the facility in question.) Larger deviations or changes to the sequence occur when unexpected events disrupt the initial schedule. Even if the managers and supervisors do not explicitly update the

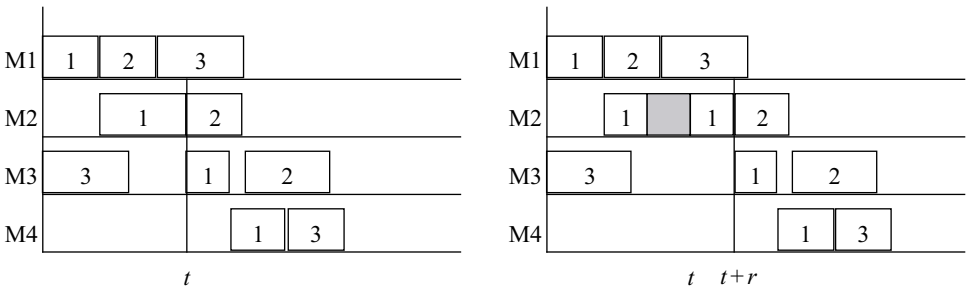

Figure 2. Using right-shift rescheduling to update a schedule

schedule, schedule repair occurs as the operators react to the disruptions, delaying tasks, or performing tasks out of order.

There are three common methods used to update (repair) a schedule that is no longer feasible due to a disruption: right shift rescheduling, regeneration, and partial rescheduling.

Right shift rescheduling postpones each remaining operation (shifting it to the right on a Gantt chart) by the amount of time needed to make the schedule feasible (Abumaizar and Svestka, 1997). For example, in the Gantt chart shown in Figure 2, if machine M2 fails while processing job 1 and the repair time requires r time units, then the completion time of Job 1 (on Machine M2) is delayed from t to $t+r$. In addition, the completion times of the remaining tasks on M2, M3, and M4 are delayed by r time units.

Partial rescheduling reschedules only the operations affected directly or indirectly by the disruption (Li, Shyu, and Adiga, 1993; Wu and Li, 1995; Olumolade and Norrie, 1996). For this reason, it is also known as affected operations rescheduling (Abumaizar and Svestka, 1997). This method preserves the initial schedule as much as possible, tending to maintain schedule stability with little nervousness. Most of the heuristics developed have considered rescheduling of affected operations only (Li, Shyu, and Adiga, 1993; Abumaizar and Svestka, 1997; Jain and Elmaraghy, 1997). Right-shift rescheduling is a special case of this method. Match-up scheduling (Bean et al., 1991) is another type of partial rescheduling.

Abumaizar and Svestka (1997) developed an algorithm for rescheduling the affected operations in a job shop with respect to efficiency and stability. They compared the system performance under the proposed affected operations method to the total rescheduling and right-shift rescheduling methods.

Bean et al. (1991) discuss a matchup scheduling procedure that repairs a production schedule when a disruption occurs. This procedure uses heuristic ordering rules to resequence all jobs scheduled before a matchup point. If the tardiness cost is too large, the matchup point is increased. If the matchup point becomes too large, the method solves an integer program or uses priority rules to reassign jobs to different machines. Their results show that matchup scheduling is an optimal approach when disruptions are infrequent enough to allow the system to get back on schedule before the next disruption. Akturk and Gorgulu (1999) present another matchup scheduling procedure that partially reschedules a modified flow shop when a machine breakdown occurs.

In OPIS, Smith et al. (1990) and Smith (1994) use a constraint-based schedule repair procedure to modify the operations in a partial schedule whose length is determined by the

conflict duration. Miyashita and Sycara (1994) describe a case-based approach for selecting a repair tactic within a constraint-based schedule repair procedure. The repair tactics include adjusting start times, swapping operations, and switching to alternative resources.

Regeneration reschedules the entire set of operations (jobs) not processed before the rescheduling point, including those not affected by the disruption (Church and Uzsoy, 1992; Wu, Storer, and Chang, 1993; Wu and Li, 1995; Olumolade and Norrie, 1996; Sabuncuoglu and Karabuk, 1999; Vieira, Herrmann, and Lin, 2000a, b). For this reason, it is also known as total or complete rescheduling (Olumolade and Norrie, 1996; Abumaizar and Svestka, 1997). Its main disadvantage is the excessive computational effort and unsatisfactory response time (Wu and Li, 1995). To overcome this problem, Bierwirth and Mattfeld (1999) present a genetic algorithm that reuses the previous solution to solve a job shop scheduling problem every time a new job arrives.

7. THE IMPACT OF RESCHEDULING POLICIES

In addition to the great deal of effort spent on rescheduling methods (as described in Section 6), another important body of work studies the impact that other aspects of the rescheduling policy have on manufacturing system performance. These other aspects include the type of events that trigger rescheduling and the rescheduling frequency. Determining the impact of a rescheduling policy on a dynamic manufacturing system requires careful study, modeling, and analysis of the specific manufacturing system.

Church and Uzsoy (1992) developed a hybrid event-driven rescheduling policy for single- and parallel-machine models with dynamic job arrivals. Their system reschedules the facility, periodically taking into account work that is already in the system. Regular events occurring between routine rescheduling are ignored until the next rescheduling moment. However, when an event is classified as an exception, immediate action should be taken, with the entire facility being rescheduled and resulting schedule implemented until the next schedule generation point. To create a schedule, the system uses the EDD rule to minimize maximum lateness. The paper also presents analytical models to bound the maximum completion time. The paper states that periodic rescheduling policies lead to near optimal performance (minimal maximum lateness) when order release is periodic. In addition, rescheduling at the arrival of a "rush" job (one with a tight due date) is useful, but more frequent rescheduling does not improve system performance significantly. Thus, if done carefully, good system performance can be maintained while reducing the rescheduling effort (the number of rescheduling events).

Vieira, Herrmann, and Lin (2000a) have studied a single-machine system and developed analytical models to estimate system performance. That work considered two rescheduling policies: periodic and event-driven based on queue size. Their results show that the analytical models can accurately predict the performance of a single-machine system operating under those rescheduling strategies. Vieira, Herrmann, and Lin (2000b) extended that study by investigating parallel machine systems, which have more complex rescheduling strategies. These papers have shown that rescheduling frequency can significantly affect the system performance (average flow time). A lower rescheduling frequency (which causes longer rescheduling periods) lowers the number of setups (reducing unproductive time wasted on setups) by grouping similar jobs but increases manufacturing cycle time and WIP. A higher rescheduling frequency allows the system to react more quickly to disruptions but may increase the number of setups.

Event-driven and periodic strategies exhibit similar performance. Rescheduling when a machine fails or becomes available after a repair decreases manufacturing cycle time slightly but increases the frequency of rescheduling.

Intuitively, it seems natural that rescheduling more often yields better performance. A number of experimental studies support this hypothesis. Farn and Muhlemann (1979) use simulation to study a single-machine system with sequence-dependent setup times. Arriving jobs are included in the schedule at the next rescheduling point, and the schedule is created using a priority rule such as first-come-first-served or SPT. They conclude that rescheduling more often leads to lower setup costs. Muhlemann, Lockett, and Farn (1982) study the dynamic job shop scheduling problem and experimentally compare different scheduling heuristics across a range of scenarios, including rescheduling period length, the number of jobs in the backlog, and the amount of uncertainty in processing times and machine failures. They also suggest that the rescheduling period affects system performance more when there is greater uncertainty and that managers need to explore the tradeoff between the cost of scheduling and the benefits of more frequent scheduling.

Bean et al. (1991) show that the matchup algorithm (which requires more job reassignments) leads to better performance (less total tardiness) than a simple pushback strategy that simply delays tasks.

According to Wu, Byeon, and Storer (1999), a robust, partial schedule leads to better system performance (less weighted tardiness) than dispatching rules. However, as processing time variability increases, dispatching rules lead to better performance. Leon, Wu, and Storer (1994) state that, as processing time variability increases, the improvement (in expected makespan and expected delay) due to robust schedules increases.

Mehta and Uzsoy (1998) state that predictive schedules (with inserted idle time) increase predictability (reduce nervousness) but do not significantly degrade system performance (maximum lateness), compared to schedules generated by ignoring possible breakdowns.

Kim and Kim (1994) considered minor and major disturbances in their scheduling system. The simulation mechanism to select a dispatching rule will be called periodically, according to a monitoring period that is a multiple of the mean operation processing time, and at major disturbances, which occur infrequently (e.g., arrival of urgent jobs and major machine breakdowns). Several values for the monitoring periods were studied. They concluded that there was an advantage to checking the system performance periodically and that too-long monitoring periods resulted in worse performance of the systems and also that too-frequent monitoring could negatively affect performance.

Sabuncuoglu and Karabuk (1999) studied the frequency of rescheduling in the multi-resource environment of a flexible manufacturing system with random machine breakdowns and processing times. For the scenario considered, they concluded that never reacting to disturbances or reacting to every disturbance do not seem to be appropriate policies. Then a moderate level of scheduling frequency is suggested to alleviate the negative effects of machine breakdowns.

One of the major objectives of Shafaei and Brunn (1999a, b) was to examine whether a more frequent rescheduling policy would always improve system performance. According to the performance measure used, they concluded that, under loose due date conditions, the performance is not particularly sensitive to changes in rescheduling interval. However, at tight due date conditions, the rescheduling interval had a much more significant effect on performance.

They also showed that frequent rescheduling becomes more effective as the level of uncertainty increases and that with the recent sharp decline in the price of computer hardware and growing increases in the capabilities of production control systems, a more frequent rescheduling policy can be more easily and economically introduced.

Although it can increase the computational effort of the rescheduling procedure (because it increases the number of jobs that are considered simultaneously), a longer rescheduling period can improve system performance through better coordination. For example, Herrmann and Delalio (2001) consider the impact of the rescheduling period on decisions regarding batching and scheduling of sheet metal punch press operations. Their results indicate that, when material is inexpensive, decreasing the scheduling frequency can significantly reduce costs because fewer setups occur and more parts are produced from inexpensive unsheared sheets. However, when material is expensive, changing the scheduling frequency does not affect costs as much.

The cost of rescheduling includes computational effort (human or computer) and disruptions to existing plans (nervousness). The rescheduling period affects the number of jobs being considered for scheduling. A longer rescheduling period means that more jobs (and tasks) will be considered in the scheduling problem. This will increase the computational effort needed to create the production schedule. Moving jobs from one scheduled machine to another may require additional material handling work. For instance, Bean et al. (1991) use the number of jobs reassigned as a measure of rescheduling cost.

8. SCHEDULING THEORY AND PRACTICE

Understanding rescheduling can address the gap between theory and practice of production scheduling. Production scheduling theory has had limited impact on practice because most scheduling results do not consider important characteristics of the environment in which scheduling occurs. In particular, researchers have not considered fully the dynamic aspects of the manufacturing system.

Solving production scheduling problems is an important technique for controlling dynamic, stochastic manufacturing systems. Viewing rescheduling as a dynamic process provides a system-level perspective of production scheduling that can put this task into proper context. Rescheduling policies identify not only when rescheduling should be done but also the objectives and constraints of the resulting scheduling problem. For example, Bean et al. (1991) present the matchup scheduling problem, which attempts to recover the original schedule as soon as possible while satisfying a constraint on allowable tardiness cost. Vieira, Herrmann, and Lin (2000a, b) study rescheduling policies that require the production schedule to minimize the number of setups and the job flow time.

Portougal and Robb (2000) discuss the gap between production scheduling theory and practice and emphasize the importance of the planning period. Their paper argues that, if job cycle times are greater than the planning period, then careful scheduling is needed to coordinate activities in multiple planning periods, and complex models are appropriate. If the cycle time is smaller, then scheduling is seldom important. The paper states that, in the latter case, the only important objective is that the resource (or production unit) completes all of the desired work in the planning period.

However, one can easily see that scheduling is critical if careless scheduling would prevent the resource (or production unit) from accomplishing this goal. In the presence of

sequence-dependent setup times, for instance, scheduling significantly affects the total time required. A poor schedule would waste valuable time doing setups. In addition, proper scheduling can support other objectives, such as minimizing the costs associated with setups.

Thus, it may be more appropriate to state that, when job cycle times are shorter than the planning period, satisfying the production target should set the constraints and objectives of the production scheduling problem. The resulting production scheduling problems may emphasize finding feasible solutions over optimization, but such problems can be extremely difficult in realistic settings.

McKay and Wiers (1999) discuss the relationship between the theory and practice of scheduling and describe three principles that explain practical production scheduling processes. First, a scheduling process generates partial solutions for partial problems. Second, a scheduling process anticipates, reacts to, and adjusts for disturbances. Third, the scheduling process is sensitive to and adjusts to the meaning of time in the production situation. All three principles support the perspective that scheduling is part of a dynamic process.

9. SUMMARY AND CONCLUSIONS

A great deal of effort has been spent developing methods to generate optimal production schedules, and countless papers discussing this topic have appeared in scholarly journals. Typically, such papers formulate scheduling as a combinatorial optimization problem. However, the scope of papers on rescheduling, a necessary part of managing a dynamic manufacturing system, varies greatly. Although all rescheduling approaches, at their core, seek to help a manufacturing system run more productively and efficiently, papers describing these approaches address a wide variety of topics. Some papers describe algorithms for generating or updating production schedules. Other papers present new rescheduling policies that specify when production schedules are generated and updated. Other papers present studies on dispatching rules, optimal control policies, or other rescheduling strategies. There are many rescheduling environments discussed by these papers.

For this reason, this paper has presented a framework for understanding rescheduling research and defined a number of terms used in rescheduling research and practice. The framework includes rescheduling environments, rescheduling strategies, rescheduling policies, and rescheduling methods.

There are two common rescheduling strategies: dynamic scheduling and predictive-reactive scheduling. Predictive-reactive scheduling includes three types of policies: periodic, event-driven, continuous, and hybrid rescheduling. Under a periodic policy, a schedule is revised (or created) periodically over time. Under an event-driven policy, rescheduling occurs when certain events occur, including machine breakdown, rush order arrival, and order cancellation. A hybrid rescheduling policy will periodically update a schedule unless a rescheduling event takes place. Dynamic, or continuous, rescheduling is a special case of event-driven rescheduling, since its approach is to reschedule the system at every rescheduling event, including, for instance, job arrival.

The three most common schedule repair methods are regeneration, partial rescheduling, and right-shift scheduling. Regeneration constructs a complete schedule by rescheduling not only the affected operations (or jobs) but also those not affected. For this reason, it is also called total rescheduling. This strategy takes more computational effort to run since more operations must be scheduled. On the other hand, better schedules can be created. It yields the most schedule

nervousness (and least stability). Partial rescheduling is also called affected operations rescheduling, since it reschedules only those operations that were affected by the disruption. This reduces the schedule nervousness (and increases stability). The right-shift method postpones the remaining operations by the amount of downtime. In some cases, right-shift might be a special case of partial rescheduling. The right-shift method yields the least schedule nervousness (and most schedule stability).

This paper did not discuss the details of the many algorithms used to generate and update production schedules. We leave such a detailed review to others.

Instead, this paper focused on the entire area of rescheduling with the hope that this will help practitioners, researchers, and students understand this body of knowledge. A comprehensive classification of papers (along the lines of Table I) is possible but beyond the scope of this paper due to the huge number of papers on scheduling manufacturing systems.

Theory and Practice. Studying rescheduling helps bridge the gap between theory and practice of production scheduling. Most scheduling results do not consider important characteristics of the dynamic environment in which scheduling occurs, which limits their usefulness. Rescheduling provides a systems view of manufacturing that includes not only material flow and resource availability but also order release and production control systems.

Modeling rescheduling. Mathematical models of dynamic, stochastic manufacturing systems can provide useful information to analysts and managers trying to design manufacturing systems. There are a wide variety of models available, including queueing network models and discrete event simulation models. Typically, however, these types of models do not explicitly represent the production control policies (e.g., rescheduling policies) that will control the system. Consequently, because these policies significantly affect system performance, the resulting system models will be inaccurate, which can lead to poor design decisions.

Because the rescheduling policy affects the performance of the manufacturing system, it needs to be considered in manufacturing system design. Rarely is the dynamic behavior of the manufacturing system considered during the design phase. When it is, more effort is spent modeling the resources in the factory and the flow of parts through the system. Little effort is spent modeling the production control scheme. This occurs because existing analytical and simulation models provide little support for rescheduling. Often, they are limited to predefined sets of dispatching rules. Although modern software for building discrete event simulation models allows an analyst to create complex models and sophisticated production control policies, building such models and conducting the necessary experiments can require a large amount of time and effort.

More research is needed to compare the performance of manufacturing systems under predictive-reactive rescheduling policies with their performance under dynamic scheduling (such as dispatching rules). This will yield additional insight into the advantages and disadvantages of rescheduling in different problem settings. This study could be done by examining analytical models (for those systems where such models exist or can be constructed) or conducting simulation studies (for more complex systems). Although there have been some studies, a comprehensive campaign is still needed. In addition, more research is needed to understand how the interactions between rescheduling policies and other production planning functions (such as capacity planning and material requirements planning) affect manufacturing system performance. Finally, this line of research could be applied to other types of dynamic, stochastic decision-making systems (such as supply chains) where planning and scheduling activities affect system behavior.

ACKNOWLEDGMENTS

The first author received support (grant number 3135/95-3) from *CAPES*, a research foundation in Brazil, to perform research in the Computer Integrated Manufacturing Laboratory at the University of Maryland. The authors thank the referees for the many useful suggestions that they made during the review process.

REFERENCES

Abumaizar, R. J. and J. A. Svestka, "Rescheduling job shops under disruptions." *Int. J. Prod. Res.*, **35**, 2065–2082 (1997).

Akkan, C., "Overtime scheduling: an application in finite-capacity real-time scheduling," *J. Oper. Res. Soc.*, **47**, 1137–1149 (1996).

Akturk, M. Selim and Elif Gorgulu, "Match-up scheduling under a machine breakdown," *Eur. J. Oper. Res.*, **112**, 81–97 (1999).

Arslan, Hasan, Hayriye Ayhan, and Tava Lennon Olsen, "Analytic models for when and how to expedite in make-to-order systems," *IIE Trans.*, **33**, 1019–1029 (2001).

Baker, K. and D. W. Peterson, "An analytical framework for evaluating rolling schedules." *Manage. Sci.*, **25**(4), 342–351 (1979).

Basnet, C. and J. H. Mize, "Scheduling and control of flexible manufacturing systems: a critical review." *Int. J. Comp. Integr. Manufact.*, **7**(6), 340–355 (1994).

Bean, James C., John R. Birge, John Mittenthal and Charles E. Noon, "Matchup scheduling with multiple resources, release dates, and disruptions," *Oper. Res.*, **39**(3), 470–483 (1991).

Bierwirth, Christian and Dirk C. Mattfeld, "Production scheduling and rescheduling with genetic algorithms," *Evolutionary Computation* **7**(1), 1–17 (1999).

Bispo, Carlos F. and Sridhar Tayur, "Managing simple re-entrant flow lines: theoretical foundation and experimental results," *IIE Trans.*, **33**, 609–623 (2001).

Black, J. T., "Manufacturing systems," in Paul M. Swamidass (ed.), *Encyclopedia of Production and Manufacturing Management*, Kluwer, Norwell, Massachusetts, 2000.

Brown, Donald E. and William T. Scherer, *Intelligent Scheduling Systems*, Kluwer Academic Publishers, Norwell, Massachusetts, 1995.

Browne, Jim (ed.), *Knowledge-Based Production Management Systems.* Elsevier Science Publishers, Amsterdam, 1989.

Buzacott, John A. and George J. Shanthikumar, *Stochastic Models of Manufacturing Systems*. Prentice Hall, Englewood Cliffs, New Jersey, 1993

Byeon, Eui-Seok, S. David Wu, and Robert H. Storer, "Decomposition heuristics for robust job shop scheduling," *IEEE Trans. Robot. Autom.*, **14**(2), 303–313 (1998).

Chacon, G. R., "Using simulation to integrate scheduling with the manufacturing execution system." *Future Fab Inter.*, 63–66 (1998).

Chase, C. and P. J. Ramadge, "On real-time scheduling policies for flexible manufacturing systems." *IEEE Trans. Autom. Control*, **37**(4), 491–496 (1992).

Church, L. K. and R. Uzsoy, "Analysis of periodic and event-driven rescheduling policies in dynamic shops." *Inter. J. Comp. Integr. Manufact.*, **5**, 153–163 (1992).

Cowling, Peter and Marcus Johansson, "Using real time information for effective dynamic scheduling," *Eur. J. Oper. Res.*, **139**(2), 230–244 (2002).

Daniels, R. L. and P. Kouvelis, "Robust scheduling to hedge against processing time uncertainty in single-stage production," *Manage. Sci.*, **41**(2), 363–376 (1995).

Dhingra, J. S., K. L. Musser, and G. L. Blankenship, "Real-time operations scheduling for flexible manufacturing systems." *Proc. 1992 Winter Simulation Conf.*, 849–855 (1992).

Dutta, A., "Reacting to scheduling exceptions in FMS environments." *IIE Trans.*, **22**(4), 300–314 (1990).

Fang, J. and Y. Xi, "A rolling horizon job shop rescheduling strategy in the dynamic environment." *Inter. J. Adv. Manufact. Tech.*, **13**, 227–232 (1997).

Farn, C. K. and A. P. Muhlemann, "The dynamic aspects of a production scheduling problem," *Inter. J. Prod. Res.*, **17**, 15–21 (1979).

Gershwin, Stanley B., *Manufacturing Systems Engineering*. PTR Prentice Hall, Englewood Cliffs, New Jersey, 1994.

Graham, R. L., E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan, "Optimization and approximation in deterministic machine scheduling: a survey," *Ann. Discrete Math.* **5**, 287–326 (1979).

Green, G. I. and L. B. Appel, "An empirical analysis of job shop dispatch rule selection," *J. Oper. Manage.*, **1**, 197–203 (1981).

Hall, Nicholas G. and Chelliah Sriskandarajah, "A survey of machine scheduling problems with blocking and no-wait in process," *Oper. Res.*, **44**(3), 510–525 (1996).

Henning, G. P. and J. Cerda, "An expert system for predictive and reactive scheduling of multiproduct batch plants." *Latin Am. App. Res.*, **25**, 187–198 (1995).
Herrmann, Jeffrey W., "A genetic algorithm for minimax optimization problems," *Proceedings of the 1999 Congress on Evolutionary Computation*, Washington, DC (July 6–9) 1999, pp. 1099–1103.
Herrmann, Jeffrey W., Chung-Yee Lee, and J. Snowdon, "A classification of static scheduling problems," in P. M. Pardalos (eds.), *Complexity in Numerical Optimization.* World Scientific, 1993.
Herrmann, Jeffrey W. and David R. Delalio, "Algorithms for sheet metal nesting," *IEEE Trans. Robot. Autom.*, **17**(2), 183–190 (2001).
Herrmann, Jeffrey W., Chung-Yee Lee, and Jim Hinchman, "Global job shop scheduling with a genetic algorithm," *Production and Operations Management*, **4**(1), 30–45 (1995).
Hopp, Wallace J. and Mark L. Spearman, *Factory Physics*, Irwin/McGraw-Hill, Boston, 1996.
Huang, Y.-G. , L. N. Kanal, and S. K. Tripathi, "Reactive scheduling for a single machine: Problem definition, analysis, and heuristic solution." *Int. J. Comput. Integr. Manufact.*, **3**(1), 6–12 (1990).
Jain, A. K. and H. A. Elmaraghy, "Production scheduling/rescheduling in flexible manufacturing ," *Int. J. Prod. Res.*, **35**, 281–309 (1997).
Jones, A., F. Riddick, and L. Rabelo, "Development of a predictive-reactive scheduler using genetic algorithms and simulation-based scheduling software." (*White paper*) (1998).
Kamoun, H. and C. Sriskandarajah, "The complexity of scheduling jobs in repetitive manufacturing systems." *Eur. J. Oper. Res.*, **70**(3), 350–364 (1993).
Kempf, Karl G., "Intelligently scheduling semiconductor wafer fabrication," in M. ZwebenM. S. Fox (eds.), *Intelligent Scheduling*, Morgan Kaufmann Publishers, San Francisco, 1994.
Kim, M. H. and Y.-D. Kim, "Simulation-based real-time scheduling in a flexible manufacturing system." *J. Manufact. Syst.*, **13**, 85–93 (1994).
Koenig, Daniel T., *Fundamentals of Shop Operations Management: Work Station Dynamics*, American Society of Mechanical Engineers, New York, 2000.
Kumar, P. R., "Scheduling manufacturing systems of re-entrant lines," in David D. Yao (eds.), *Stochastic Modeling and Analysis of Manufacturing Systems*, Springer-Verlag, New York, 1994, pp. 325–360.
Lee, Tae-Eog and Marc E. Posner, "Performance measures and schedules in periodic job shops," *Oper. Res.*, **45**(1), 72–91 (1997).
Leon, V. Jorge, S. David Wu, and Robert H. Storer, "Robustness measures and robust scheduling for job shops," *IIE Trans.*, **26**(5), 32–43 (1994).
Li, R.-K., Y.-T. Shyu, and S. Adiga, "A heuristic rescheduling algorithm for computer-based production scheduling systems." *Int. J. Prod. Res.*, **31**, 1815–1826 (1993).
Liu, J. and B. L. MacCarthy, "The classification of FMS scheduling problems," *Int. J. Prod. Res.*, **34**(3), 647–656 (1996).
Markowitz, David M. and Lawrence M. Wein, "Heavy traffic analysis of dynamic cyclic policies: a unified treatment of the single machine scheduling problem," *Oper. Res.*, **49**(2), 246–270 (2001).
Matsuo, Hirofumi, "Cyclic scheduling problems in the two-machine permutation flow shop: complexity, worst-case, and average-case analysis," *Naval Res. Logist.*, **37**, 679–694 (1990).
McKay, Kenneth N. and Vincent C. S. Wiers, "Unifying the theory and practice of production scheduling," *J. Manufact. Syst.*, **18**(4), 241–255 (1999).
McPherson, R. F. and K. P. White Jr., "Periodic flow line scheduling," *Int. J. Prod. Res.*, **36**(1), 51–73 (1998).
Mehta, S. V. and R. M. Uzsoy, "Predictable scheduling of a job shop subject to breakdowns." *IEEE Trans. Robot. Autom.*, **14**, 365–378 (1998).
Miyashita, Kazuo and Katia Sycara, "Adaptive case-based control of schedule revision," in Monte, ZwebenMark S. Fox (eds.), *Intelligent Scheduling,* pp. 29–66, Morgan Kaufmann Publishers, San Francisco, 1994.
Muhlemann, A. P., A. G. Lockett, and C.-K. Farn, "Job shop scheduling heuristics and frequency of scheduling," *Int. J. Prod. Res.*, **20**(2), 227–241 (1982).
Nof, Shimon Y. and Grant F. Hank, "Adaptive/predictive scheduling: review and a general framework," *Prod. Planning Control*, **2**(4), 298–312 (1991).
O'Donovan, Ronan, Reha Uzsoy, and Kenneth N. McKay, "Predictable scheduling of a single machine with breakdowns and sensitive jobs," *Int. J. Prod. Res.*, **37**(18), 4217–4233 (1999).
Olumolade, M. O. and D. H. Norrie, "Reactive scheduling system for cellular manufacturing with failure-prone machines." *Int. J. Comput. Integr. Manufact.*, **9**(2), 131–144 (1996).
Ovacik, I. M. and R. Uzsoy, "Rolling horizon algorithms for a single-machine dynamic scheduling problem with sequence-dependent setup times." *Int. J. Prod. Res.*, **32**(6), 1243–1263 (1994).
Ovacik, I. M. and R. Uzsoy, "Rolling horizon procedures for dynamic parallel machine scheduling with sequence-dependent setup times," *Int. J. Prod. Res.*, **33**(11), 3173–3192 (1995).
Panwalkar, S. S. and W. Iskander, "A survey of scheduling rules," *Oper. Res.*, **25**, 45–61 (1977).
Peng, C. and F. F. Chen, "A simulation based ordinal optimization approach to real-time control and scheduling of flexible manufacturing systems," *Working paper submitted for presentation at NAMRC XXVI (May 1998) and publication at Transactions of NAMRI/SME*, 1998.

Perkins, J. R. and P. R. Kumar, "Stable, distributed, real-time scheduling of flexible manufacturing/assembly/disassembly systems." *IEEE Trans. Autom. Control*, **34**(2), 139–148 (1989).

Pinedo, Michael, Scheduling: Theory, Algorithms, and Systems Prentice Hall, Englewood Cliffs, New Jersey, 1995.

Pinedo, Michael and Xiuli Chao, *Operations Scheduling with Applications in Manufacturing and Services*, Irwin McGraw Hill, Boston, 1999.

Portougal, Victor and David J. Robb, "Production scheduling theory: just where is it applicable?" *INTERFACES*, **30**(6), 64–76 (2000).

Prietula, Michael J., Hsu Wen-Ling, Ow Peng Si, and Gerald L. Thompson, "MacMerl: mixed-initiative scheduling with coincident problem spaces," in M. Zweben and M.S. Fox (eds.), *Intelligent Scheduling*, Morgan Kaufmann Publishers, San Francisco, 1994.

Roundy, Robin, "Cyclic schedules for job shops with identical jobs," *Math. Oper. Res.*, **17**(4), 842–865 (1992).

Sabuncuoglu, I. and S. Karabuk, "Rescheduling frequency in an FMS with uncertain processing times and unreliable machines." *J. Manufact. Syst.*, **18**(4), 268–283 (1999).

Shafaei, R. and P. Brunn, "Workshop scheduling using practical (inaccurate) data—Part 1: the performance of heuristic scheduling rules in a dynamic job shop environment using a rolling time horizon approach." *Int. J. Prod. Res.*, **37**(17), 3913–3925 (1999a).

Shafaei, R. and P. Brunn, "Workshop scheduling using practical (inaccurate) data—Part 2: an investigation of the robustness of scheduling rules in a dynamic and stochastic environment." *Int. J. Prod. Res.*, **37**(18), 4105–4117 (1999b).

Smith, Stephen F., "OPIS: a methodology and architecture for reactive scheduling," in Monte Zweben and Mark S. Fox (eds.), *Intelligent Scheduling,* pp. 29–66, Morgan Kaufmann Publishers, San Francisco, 1994.

Smith, Stephen F., Peng Si Ow, Jean-Yves Potvin, Nicola Muscettola, and Dirk C. Matthys, "An integrated framework for generating and revising factory schedules," *J. Oper. Res. Soc.*, **41**(6), 539–552 (1990).

Sun, D. and L. Lin, "A dynamic job shop scheduling framework: a backward approach," *Int. J. Prod. Res.*, **32**(4), 967–985 (1994).

Szelke, E. and R. Kerr, "Knowledge-based reactive scheduling." *Production Planning and Control*, **5**(2), 124–145 (1994).

Tabe, T. and G. Salvendy, "Toward a hybrid intelligent system for scheduling and rescheduling of FMS," *Int. J. Comput. Integr. Manufact.*, **1**(3), 154–164 (1988).

Vieira, Guilherme E., Jeffrey W. Herrmann, and Edward Lin, "Analytical models to predict the performance of a single-machine system under periodic and event-driven rescheduling strategies," *Int. J. Prod. Res.*, **38**(8), 1899–1915 (2000a).

Vieira, Guilherme E., Jeffrey W. Herrmann, and Edward Lin, "Predicting the performance of rescheduling strategies for parallel machine systems," *J. Manufact. Syst.*, **19**(4), 256–266 (2000b).

Vollmann, Thomas E., William L. Berry, and Clay Whybark, *Manufacturing Planning and Control Systems* (4th edn) Irwin/McGraw-Hill, New York, 1997.

Watatani, Y. and S. Fujii, A study on rescheduling policy in production system. JAPAN/USA Symp. Flexible Automation ASMe, 1992.

Wu, S. David, Eui-Seok Byeon, and Robert H. Storer, "A graph-theoretic decomposition of the job shop scheduling problem to achieve scheduling robustness," *Oper. Res.*, **47**(1), 113–124 (1999).

Wu, H.-H. and R. K. Li, "A new rescheduling method for computer based scheduling systems." *Int. J. Prod. Res.*, **33**(8), 2097–2110 (1995).

Wu, S. D., R. H. Storer and P.-C. Chang, "One-machine rescheduling heuristics with efficiency and stability as criteria." *Comput. Oper. Res.*, **20**, 1–14 (1993).

Yamamoto, M. and S. Y. Nof, Scheduling/rescheduling in the manufacturing operating system environment. *Int. J. Prod. Res.*, **23**(4), 705–722 (1985).

Zweben, Monte and Mark S. Fox, *Intelligent Scheduling*, Morgan Kaufmann Publishers, San Francisco, 1994.