PRODUCTION AND OPERATIONS MANAGEMENT
Vol. 4, No. 1, Winter 1995
Printed in U.S.A.

GLOBAL JOB SHOP SCHEDULING WITH A GENETIC ALGORITHM*

JEFFREY W. HERRMANN, CHUNG-YEE LEE, AND JIM HINCHMAN
Department of Mechanical Engineering, University of Maryland, College Park, MD 20742, *USA*
Department of Industrial and Systems Engineering, University of Florida, Gainesville, Florida 32611, *USA*
Harris Semiconductor Corporation, Melbourne, Florida 32902, *USA*

This paper describes a global job shop scheduling procedure that uses a genetic algorithm to find a good schedule. Unlike previously considered algorithms, this procedure has been implemented in the scheduling system for a manufacturing facility and has led to improved scheduling. This facility is a semiconductor test area. The test area is a job shop and has sequence-dependent setup times at some operations. The concern of management is to meet their customer due dates and to increase throughput. This requires the coordination of many resources, a task beyond the ability of simple dispatching rules. We discuss a centralized procedure that can find a good schedule through the use of a detailed scheduling model and a genetic algorithm that searches over combinations of dispatching rules. We discuss our effort in developing a system that models the shop, creates schedules for the test area personnel, and makes a number of contributions to test area management.

(JOB SHOP SCHEDULING; GENETIC ALGORITHMS; SEMICONDUCTOR MANUFACTURING)

1. Introduction

In many areas of manufacturing, the ability of a facility to meet its objectives depends on the close coordination of resources. This coordination must occur on different levels: capacity planning, release planning, and lot dispatching. Many researchers have developed approaches for the first two levels. In this paper, we consider the very difficult scheduling problems that exist at the third level. Outside of shop floor control systems that have implemented simple dispatching rules, no successful commercial packages are available for good global scheduling. However, increases in computing power mean that such a system is now feasible.

We are concerned with the effective scheduling of a semiconductor test area, which can be classified as a job shop environment. Effective scheduling can lead to improvements in throughput, efficiency, customer satisfaction (measured by meeting due dates), and other performance measures. A number of characteristics make the semiconductor test area difficult to control. These include the conflicting goals of management (balancing increased throughput against meeting due dates), work centers with sequence-dependent

* Received August 1993; revision received June 1994; accepted January 1995.

1059-1478/95/0401/030$1.25

setup times, operations where multiple lots can be processed simultaneously, and timing relationships between operations.

This paper describes a global job shop scheduling system developed for semiconductor testing. We will examine the needs of the semiconductor test area, discuss their requirements for a scheduling system, and present our approach to the job shop scheduling problem. While implemented in this particular test area, the approach could be applied to other manufacturing facilities.

The primary function of the global job shop scheduling system is to use information about the current status of the shop, the jobs to be manufactured, and the production process in order to create a schedule of activities for each work center in the shop over a fixed time period. By using a centralized procedure with global information, the system can search for a schedule better than that computed by making nearsighted, local decisions.

The system finds good solutions with a genetic algorithm, a type of heuristic search. One interesting characteristic of this search is that it looks for an efficient schedule by testing different combinations of dispatching rules with a finite capacity scheduling simulation. The use of dispatching rules is an effective local procedure to find a job shop schedule; such techniques do not, however, use any global information. The search over the combinations of dispatching rules overcomes this deficiency. While global scheduling techniques have been proposed before, the important contributions of our work are the development of a genetic algorithm for job shop scheduling and the implementation of a scheduling system that uses this genetic algorithm to discover better schedules.

The next section of this paper will discuss some background and related research on job shop scheduling. The genetic algorithm for global scheduling is introduced in Section 3, and a small example of the search space is presented. Section 4 describes the scheduling problem in the semiconductor test area under consideration and the global job shop scheduling system developed. We summarize the paper in Section 5 and describe how a similar system may be useful in other manufacturing environments.

2. Job Shop Scheduling

The problem of coordinating resources to ensure efficient manufacturing is a difficult problem. When the shop is producing parts that have the same route and require short setups, a just-in-time (JIT) philosophy may result in efficient scheduling through a pull system. In many situations, however, the manufacturing process is much more complicated (more significant changeovers and more variety in the product routings), and finding an optimal or near-optimal schedule is an impossible task.

The traditional method of controlling job shops is the use of dispatching rules. A dispatching rule is a sequencing policy that orders the jobs waiting for processing at a machine; the ordering depends upon the particular dispatching rule used. Common rules include the shortest processing time (SPT) and earliest due date (EDD) rules.

While such rules have been the subject of much research, standard dispatching rules have a narrow perspective on the scheduling problem, since they ignore information about other jobs and other resources in the shop. More advanced look-ahead and look-behind rules attempt to include more information, but their reach is still limited. Effective job shop scheduling depends upon the interaction of a number of factors. The complexity of these interactions makes the development of a global scheduling system a difficult task.

A number of researchers have looked at semiconductor manufacturing at all levels, from production planning to scheduling. Approaches to shop floor control include lot release policies, dispatching rules, deterministic scheduling, control-theoretic approaches, knowledge-based approaches, and simulation. Uzsoy, Lee, and Martin-Vega (1992a, 1993) review a substantial number of papers that consider these approaches to semiconductor scheduling.

The research into semiconductor manufacturing includes corporate-wide production planning that uses a rate-based model of production and linear programming (Hackman and Leachman 1989; Leachman 1993). Such a global planning system has been developed under the name IMPReSS at Harris Semiconductor. Hung and Leachman (1992) have developed an iterative method that uses a linear program and a discrete-event simulation to develop long-term production plans for a wafer fabrication.

While most of the research in semiconductor scheduling has concentrated on the fabrication of semiconductor wafers, a number of other papers have addressed the problems of semiconductor test. These include Lee, Uzsoy, and Martin-Vega (1992), and Uzsoy, Lee, Martin-Vega, and Hinchman (1991a); Uzsoy, Martin-Vega, Lee, and Leonard (1991b); and Uzsoy, Lee, and Martin-Vega (1992b). Lee, Martin-Vega, Uzsoy, and Hinchman (1993) report on the implementation of a decision support system for the dispatching of lots in a semiconductor test area. Our work is the natural extension of this system.

Previous approaches to the production of detailed shop schedules for planning and shop floor control include expert systems like ISIS (Fox and Smith 1984), OPIS (Smith, Fox, and Ow 1986; Ow and Smith 1988), MICRO-BOSS (Sadeh 1991), and OPAL (Bensana, Bel, and Dubois 1988); cost-based procedures such as OPT (Optimized Production Technology, reviewed by a number of authors, including Jacobs 1984), Faaland and Schmitt (1993), and the bottleneck dynamics of SCHED-STAR (Morton, Lawrence, Rajagopolan, and Kekre 1988); simulation (Leachman and Sohoni; Najmi and Lozinski); and leitstands (scheduling decision support systems described by Adelsberger and Kanet 1991). Recently, Adler et al. (1993) have described the implementation of a bottleneck-based scheduling support system for a paper bag production flow shop. While these approaches have been developed for a number of different manufacturing processes, the complications of the semiconductor test area forced us to consider a new design.

Job shop scheduling, as one of the most difficult scheduling problems, has attracted a lot of attention from researchers. Techniques such as the shifting bottleneck algorithm (Adams, Balas, and Zawack 1988) or bottleneck dynamics (see Morton 1993, for example) concentrate on solving the problem one machine at a time. More work has gone into the development and evaluation of various dispatching rules. Panwalkar and Iskander (1977) present a list of over 100 rules. Recent studies include Fry, Philipoom, and Blackstone (1988), Vepsalainen and Morton (1988), and Bhaskaran and Pinedo (1991).

More sophisticated *look-ahead* and *look-behind* rules have also been discussed. Look-behind rules (called *x-dispatch* by Morton 1992) consider the jobs that will be arriving soon from upstream machines. Look-ahead rules consider information about the downstream machines. This includes the work-in-next-queue and the number-in-next-queue rules of Panwalkar and Iskander (1977), bottleneck starvation avoidance (Glassey and Petrakian 1989), and lot release policies that look-ahead to the bottleneck (Glassey and Resende 1988; Leachman, Solorzano, and Glassey 1988; Wein 1988). Look-ahead and look-behind scheduling problems have been studied in Herrmann and Lee (1992) and Lee and Herrmann (1993).

Heuristic searches have also been developed for job shop scheduling. Van Laarhoven, Aarts, and Lenstra (1988) study the disjunctive graphs associated with a job shop scheduling problems and develop a simulated annealing approach to minimize the makespan. Matsuo, Suh, and Sullivan (1988) use a different type of simulated annealing for the same problem. Barnes and Chambers (1991) use a tabu search to find good job shop schedules. Genetic algorithms (described in detail in the next section) have been suggested for job shop scheduling by Davis (1985), Storer, Wu, and Vaccari (1991), Fox and McMahon (1990), and Nakano and Yamada (1991). These researchers have concentrated on the classical job shop problem of minimizing makespan. In particular, the more well-known procedures use the disjunctive graph to represent the problem. Unfortunately,

manufacturing environments like semiconductor test areas often have more complicated problems that require more complicated scheduling models.

3. A Genetic Algorithm for Job Shop Scheduling

One approach to difficult scheduling problems such as job shop scheduling is local search, an iterative procedure that moves from solution to solution until it finds a local optimum. *Smart-and-lucky searches* (or heuristic searches, or probabilistic search heuristics) attempt to overcome the primary problem of these simple searches: convergence to local optima. These more complex searches are smart enough to escape from most local optima; they still must be lucky, however, in order to find the global optimum.

Common smart-and-lucky searches include simulated annealing, tabu searches, and genetic algorithms. Simulated annealing was developed independently by Kirkpatrick, Gelatt, and Vecchi (1983) and Cerny (1985). Tabu search can be traced to a paper by Glover (1977). Holland (1975) developed the ideas of genetic algorithms. Recent books on genetic algorithms include Davis (1991) and Goldberg (1989).

3.1. *The Basics of Genetic Algorithms*

The genetic algorithm (sometimes abbreviated as GA) is a procedure that mimics the adaptation that nature uses to find an optimal state. In a genetic algorithm, solutions are represented as strings (chromosomes) of alleles, and the search performs operations on the population of solutions. Liepins and Hilliard (1989) identify these operations as the evaluation of individual fitness, the formation of a gene pool, and the recombination and mutation of genes to form a new population. After a period of time, good strings dominate the population, providing an optimal (or near-optimal) solution.

The solution string may be a sequence of binary bits or a permutation of objects. The fitness of a solution is its objective function evaluation or ranking. In forming the gene pool, the algorithm selects, through some random process, those solutions that are more fit. The recombination is some type of *crossover,* where offspring solutions are created with the characteristics of their two parents. It is this powerful crossover mechanism that makes genetic algorithms special. A *mutation* operation is used to maintain diversity by randomly changing some solutions.

The power of a genetic algorithm lies in the fact that each individual in the population carries a number of patterns (schema) in its chromosome. These patterns are shared by other individuals and form the building blocks of good solutions. The more schema that an individual shares with an optimal solution, the more likely it is for that individual to have a good fitness and to be selected to parent offspring. A genetic algorithm works because good (short) schema survive exponentially, leading to high-quality solutions. Moreover, the population of solutions provides an implicit parallelism that allows many areas of the search space to be explored at once.

A genetic algorithm, however, may have difficulty finding good solutions if the strings are inappropriate representations of solutions, the strings exclude important problem information (such as constraints), or the algorithm converges to a local optimum. The first issue is the most important for scheduling problems: representation is difficult for combinatorial problems because standard crossovers on permutations may lead to infeasible strings.

3.2. *The Heuristic Space*

The idea of searching heuristic and problem spaces was reported by Storer, Wu, and Vaccari (1990), who examine the general job shop scheduling problem. They define a heuristic space composed of vectors of dispatching rules. Each rule in a particular vector is used for a fixed number of dispatching decisions, regardless of the machine being

TABLE 1
Due Date and Route of Each Job

Job	Due Date	Sequence of Machines to Visit
J_1	8	Machine 1, machine 2, machine 3
J_2	12	Machine 1, machine 2
J_3	17	Machine 2, machine 1, machine 3
J_4	15	Machine 2, machine 3

scheduled, in order to generate a schedule. That is, all of the machines use the same dispatching rule at the same time until the next rule replaces it. (Papers that investigate searches over problem spaces for scheduling problems include Herrmann and Lee 1993; Wu, Storer, and Chang 1993; Bean 1994).

The approach presented here has a different perspective. In the dynamic job shop scheduling environment, the number of operations to be scheduled is unknown. Thus, it is impossible to divide the scheduling horizon by allocating to each rule a fixed number of operations. Instead, a separate dispatching rule is assigned to each machine. The system can evaluate the combination of dispatching rules (which we call a *policy*) by measuring

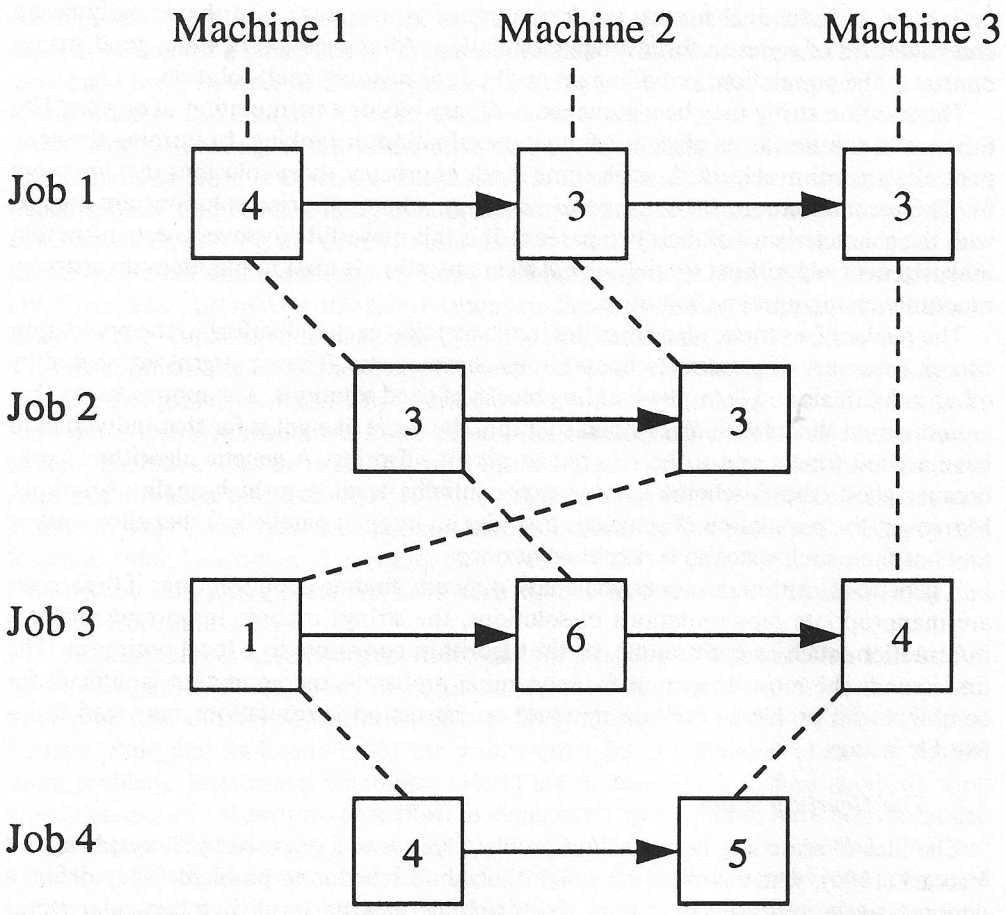

FIGURE 1. Task Processing Times for Each Job. Tasks Grouped by a Dotted Line are on the Same Machine.

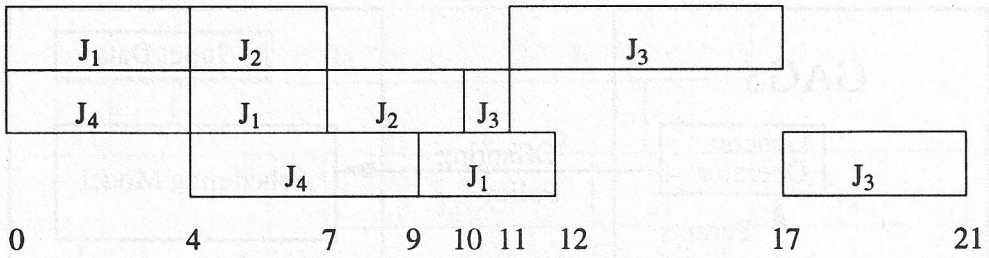

FIGURE 2. Policy 1: [EDD, EDD, EDD]. Makespan = 21.

the performance of the schedule that is created by using this policy. Changing the policy by modifying the dispatching rule on one or more machines changes the schedule created.

In order to demonstrate this idea, consider the following four-job, three-machine problem (Table 1 and Figure 1) and two policies used to dispatch the jobs waiting for processing.

Policy 1 = [EDD, EDD, EDD]. Under this policy, the tasks waiting for processing at M_1 are sequenced by earliest due date first. Thus, when both J_1 and J_2 are at M_1, J_1 is sequenced first since its due date is 8 (see Figure 2).

The dispatching rule for M_2 is EDD, so the tasks are sequenced by the job due date. When J_3 and J_4 are at M_2, J_4 is selected. After this task is completed, J_1 has arrived and its due date is also smaller than J_3. J_3 is delayed again when J_1 finishes and J_2 simultaneously arrives.

Policy 2 is [SPT, SPT, SPT], which leads to a different ordering of the jobs (see Figure 3). On M_1 the job J_2 has the shortest task processing time and is thus preferred. On M_2 the job J_3 has the shorter task processing time and is scheduled first.

Obviously, the application of a different policy creates a different schedule with a different objective function value on any objective one could wish to measure (makespan, defined as the maximum job completion time, was chosen here only as an example).

These policies can be easily manipulated by a genetic algorithm. Traditional genetic algorithms must be modified for machine scheduling problems since the components of the strings (the jobs in the sequence) are not independent of each other. The heuristic space described above, however, consists of vectors that have independent elements. That is, the dispatching rule for the first machine does not affect what values the other elements can have. Thus, a crossover operation that breaks two strings (vectors) and joins the separate pieces yields offspring that are valid points in the search space.

3.3. *A Genetic Algorithm for Global Scheduling*

The Genetic Algorithm for Global Scheduling (GAGS) is the engine that finds good schedules for the given scheduling problem. GAGS consists of two primary components:

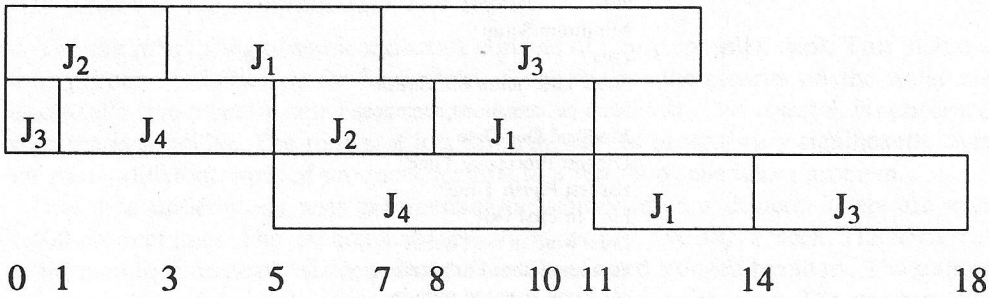

FIGURE 3. Policy 2: [SPT, SPT, SPT]. Makespan = 18.

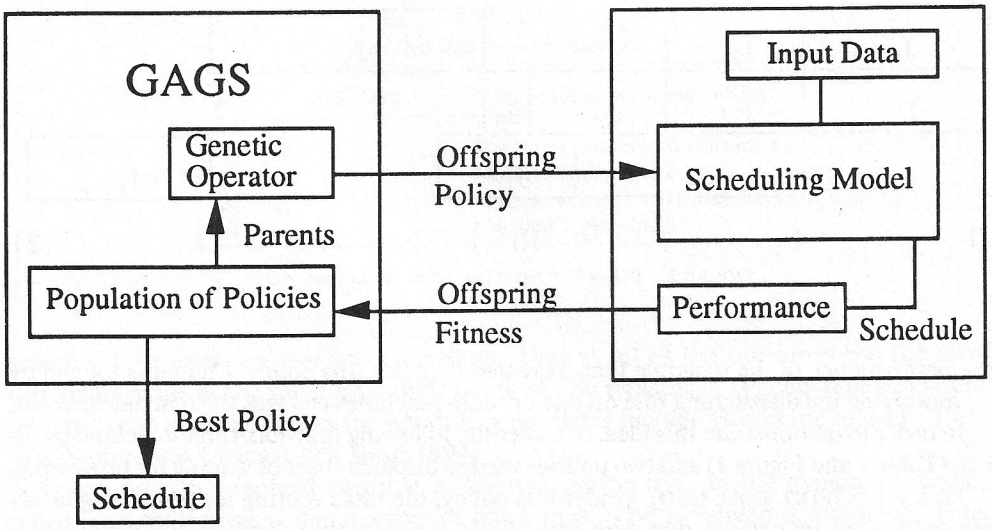

FIGURE 4. GAGS-Scheduling Model Interface.

the genetic search and the model of the shop floor. The interaction between these two functions is outlined in Figure 4. The genetic algorithm starts with a number of policies (see Table 2 for a list of the dispatching rules that were used in the global scheduling procedure). The fitness of each policy is the quality of the schedule created if the jobs in the shop are dispatched according to the policy. The model of the shop floor is employed to build this schedule from the set of shop, job, and process information.

The genetic algorithm creates new offspring policies by combining two policies in the population (crossover) or changing a solitary parent (mutation). Each offspring is evaluated using the scheduling model. For instance, consider again the above three-machine problem and the two policies [EDD, EDD, EDD] and [SPT, SPT, SPT]. Their fitness evaluations (if we use the makespan objective function) were 21 and 18. If we split each policy after the first rule and link the beginning of the first policy with the remainder of the second policy, we create the offspring policy [EDD, SPT, SPT], which the reader can confirm creates a schedule with a makespan of 17 (see Figure 5).

The quality of the schedules produced by the genetic algorithm was tested by using the search to find solutions for a few example problems. For a scheduling problem based

TABLE 2
List of Dispatching Rules Considered

SPT
Minimum Setup
EDD
Shop: Late-Setup Match-EDD
Slack per remaining operation
Modified Due Date
Longest Processing Time
Earliest Finish Time
First In First Out
Least work in next queue
EDD Look-ahead to burn-in
SPT Look-ahead to burn-in
Look-behind from burn-in

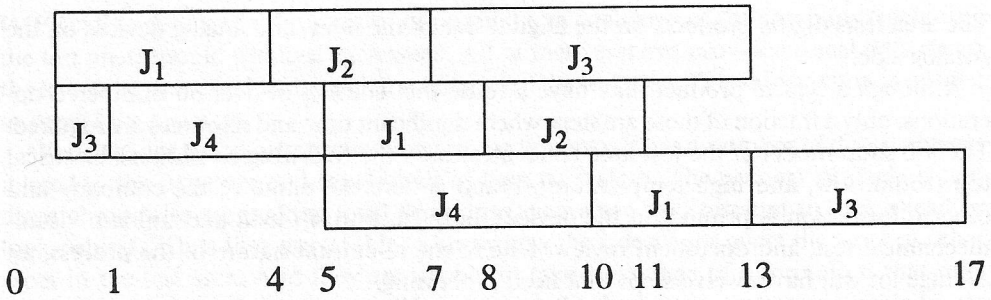

FIGURE 5. Policy 3: [EDD, SPT, SPT]. Makespan = 17.

on the test area under study, the search was able to find schedules with 12% less total flowtime than the average schedule produced by a standard dispatching rule. On a classic 10-job by 10-machine problem, the search found schedules with 16% less makespan than schedules generated by dispatching rules. These rules included SPT, LPT, FIFO, EDD, Shortest Remaining Processing Time, Slack per Remaining Operation, Modified Due Dates, and Work in Next Queue. Since the genetic algorithm examines a number of different dispatching rule combinations, it is clear that it will be able to find, for any objective function, a better schedule than one generated by a fixed sequencing policy.

In this procedure, the model of the shop floor is the problem to be solved, and the genetic algorithm provides different heuristics. The system depends upon this model, therefore: only with a valid model can the scheduling system create an implementable plan. In this work, the model is a deterministic simulation of the shop. It uses as input a set of resources that correspond to the equipment and staffing of the test area, a set of jobs that correspond to the lots that need processing in the test area, and expert knowledge about the production processes. The dispatching rules in the policy provided by the genetic algorithm sequence the jobs at each resource. Events in the simulation correspond to resources beginning work on a task and resources finishing tasks. Other events are added as necessary to control the simulation of the production process.

Note that the search is not affected by the complexity of the scheduling problem being solved. The genetic algorithm can find solutions to classical problems and to problems with previously unconsidered characteristics. In addition, it can take advantage of complexity by including dispatching rules designed for use in that environment.

4. Global Job Shop Scheduling

This section discusses the model of the manufacturing process and the scheduling needs of the semiconductor test area under investigation. The design, implementation, and contributions of the global job shop scheduling system are also presented.

4.1. *Modeling the Semiconductor Test Process*

The manufacturing of semiconductors consists of many complex steps. This includes four primary activities: wafer fabrication, probe (where the circuits on the wafer are electrically tested before being separated), assembly, and test. This research is concerned with the last facility. The routes of lots through the test process vary significantly over the many different types of products, leading to a job shop scheduling problem.

The area under study tests commercial-use semiconductor devices. There are over 1,400 product lines. The test area has three shifts per day, five days a week. The resources in the area include nearly sixteen electrical test heads and a dozen branders. The staffing on a normal shift consists of nearly fifteen personnel in eight areas. The product mix changes continuously, and staffing and machine resources often change to reflect this.

The area tests digital products on the Digital side of the floor, and analog devices on the Analog side.

Although a typical product may have a route that consists of over 30 numbered operations, only a fraction of those are steps where significant time and resources are required. The job shop model of the test area concentrates on the following operations: electrical test (room, low, and high temperature), brand (where the name of the company and other information is printed on the device), burn-in, burn-in load and unload, visual-mechanical test, and document review. Due to the re-entrant nature of the process, an average lot will have twelve steps that need processing.

Most of the operations in the semiconductor test area are electrical or physical tests, performed in an automated fashion or by hand. During these tests, each device in a lot must be examined. Thus, the processing times depend directly upon the size of the lot. Also, the products in the test area have different package types, which require different resources at the electrical tester in order to examine the devices in the lot.

In the burn-in area, an operator loads the devices in the lot onto a specific type of burn-in board; these boards are placed in an oven and stressed for a specified period of time. (An oven may hold more than one lot simultaneously.) After this period, the devices must be unloaded from the boards and then undergo electrical testing within a 96-hour window.

In our model of the test area, a work center exists for each tester, machine, or operator that needs to be scheduled. The burn-in ovens and boards are modeled separately. Each work center has a list of the operations that can be performed at the station. Distinct resources that perform the same operations are modeled as separate stations, each with a queue of jobs that next need processing at that station. These queues are sequenced by the dispatching rule for that station. Thus, different combinations of dispatching rules create different schedules.

As Lee, Martin-Vega, Uzsoy, and Hinchman (1993) mention, the test area has five features that distinguish it from classical job shop scheduling:

1. Sequence-dependent setup times and re-entrant product flows,
2. Machines with different scheduling characteristics,
3. Complex interactions between machines,
4. Dynamic production environment, and
5. Multiple, conflicting objectives.

These characteristics make effective scheduling of the semiconductor test area a difficult task. They also encourage us to design a global scheduling system that can search for good solutions without being obstructed by complexity.

4.2. *Scheduling System Background*

The test area was using a set of dispatching stations to sequence the lots awaiting processing at each work center at the beginning of the shift. (The development of this system is described in Lee, Martin-Vega, Uzsoy, and Hinchman 1993.) During the shift, this ordered list of lots was updated by using the dispatch station to resequence the lots in the queue, since more lots may have arrived. The dispatching stations and rules are part of a software module called Short-Interval Scheduling (SIS).

SIS is a component of WORKSTREAM, a product of Consilium, Inc. (Consilium 1988). WORKSTREAM is the test area's computer-integrated manufacturing (CIM) system and is implemented on the corporate VAX mainframe. Transactions such as processing a lot or beginning a setup are logged into WORKSTREAM, which maintains information about the current status of each lot and each machine. This information is used by SIS to sequence the lots waiting for processing.

In addition to WORKSTREAM, the company uses higher-level systems such as Activities Planning and Dispatching (AP/D) to match the production lots to customer orders and

IMPReSS to determine the amount of product that each area of the company (including the test area) should produce each week. All of these systems provide critical data about the lots to be processed and the characteristics of the test area. This information is needed to model the test area.

While SIS has led to better scheduling in the test area, it has a number of disadvantages related to the structure and capabilities of the CIM system. The primary obstacle is that dispatching rules are making local decisions (even when they attempt to look-ahead or look-behind). Thus they are unable to know how their decisions affect the work at other areas in the test area. And they are unable to take advantage of information that may lead to better dispatching decisions. In addition, the dispatching rules are unable to use information about processing and setup times.

The test area planners need to efficiently schedule the shop floor, to know when operations are going to be completed, and to see how the assignment of personnel affects test area performance. The test area managers need to measure operator performance, to meet customer due dates, and to test as much product as possible.

To meet these requirements, a scheduling system has to use global information about this complex production process and to search for a better schedule. Therefore, we developed the genetic algorithm for global scheduling (GAGS), which is described in Section 3.

4.3. *Scheduling System Design*

The global scheduling system includes a simulation model of the test floor and an optimization procedure that can search for schedules using a genetic algorithm (GAGS). The search over combinations of dispatching rules finds a schedule that is better than any schedule that a predefined set of dispatching rules could create. This schedule specifies the order in which the lots should be processed and the times at which the operations should be done.

The system can be used to react to unforeseen events that might occur during the course of a shift by including new information and producing a new schedule; for example, a machine may fail, or an important lot may arrive. The system also serves in a decision support manner, allowing the planners to determine the effects that changing resources has upon the schedule. Through the use of reasonably accurate processing time estimates, the schedule can be used as a target for the shift; the work completed by the shift personnel can be compared to the work on the schedule.

The foundation for the system is the test area's extensive CIM system, which provides the information necessary to model the test area. Finally, the global scheduling system is under the control of production planners, who use the computing power of the system to evaluate alternative plans and to create a detailed schedule for the shop floor.

The most important component of a global job shop scheduling system is the model of the shop floor. Only with a valid model can a scheduling system create plans that match reality. For our project, the model was a deterministic simulation of the test area. It uses as input a set of resources that correspond to the equipment and staffing of the test area, a set of jobs that correspond to the lots that need processing in the test area, and a set of dispatching rules that sequence the jobs at the resources. Thus, the model creates a schedule for the current scenario.

4.4. *Information Requirements*

The scheduling system makes use of data from a number of sources inside and outside the test area's CIM system (see Figure 6). Three general types of files can be described: data extracted from the CIM, data collected by the project team, and data entered by the user. The first category includes data that is fairly stable and data that constantly changes. Stable data includes information about the product routes. This Product-Operation (PROP)

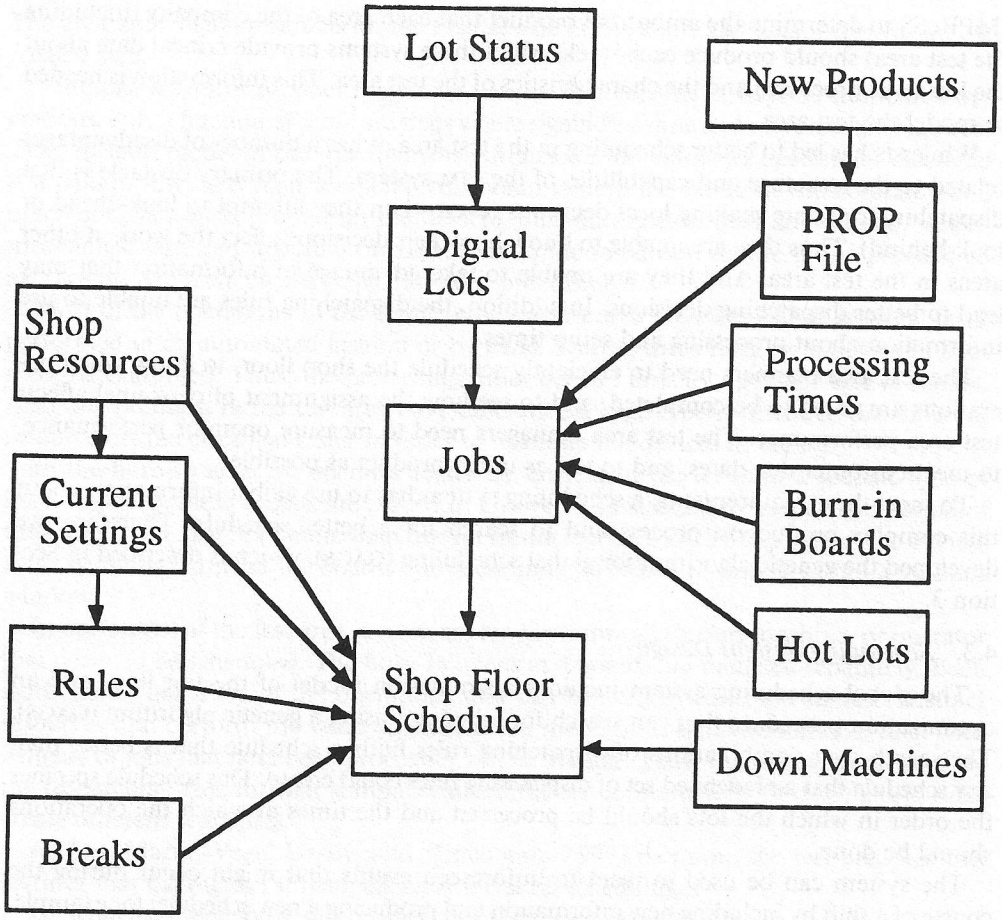

FIGURE 6. Data Structures.

file matches each product to the operations that need to be performed. This file is updated weekly as new products are added to the product mix. Other files of extracted information list package types, burn-in board requirements, and tester requirements.

The dynamic data is primarily the lot status information. The status of a lot changes as it undergoes different operations. Before each shift the CIM system takes a snapshot of the status for each lot in the test area. The scheduling system extracts from this report lot status information for each lot in the Digital side of the area. The status of a lot continues to change during the day, and these changes are noted on the next report.

The project team also collected data on processing times and test area resources. The collection of processing times will be discussed in Section 4.6. The resources file matches each type of resource (tester, brander) to the operations that it can process.

Finally, the user controls other information needed for the scheduling system, including the current test area resources (how many of each type), the dispatching rules to use, the schedule of employee breaks, the status of any down machines, and the arrival of any new lots which must be expedited ("hot lots").

4.5. *Implementation of Global Scheduling*

After creating the scheduling model and the genetic algorithm off-line, we installed the basic programs on the corporate mainframe. We began testing the system and de-

veloping the utilities that collect the necessary job, shop, and process information. Working closely with the test area personnel, we began to run schedules each day. Feedback from these initial schedules led to improvements in all parts of the system. A technology transfer session formally introduced the system capabilities to the test area personnel, and soon they began to run the system themselves. The project team created user interface programs and documentation so that the test area would be able to use, understand, and maintain the system.

The scheduler performs shift scheduling for the Digital side of the test area. Output from the system includes the shift schedule, which lists each work center, the lots to be scheduled, and the approximate start and finish times for each operation. Also available are post-shift performance reports that compare the production of the test area to the schedule.

User interface programs make it easy for the users to collect and modify data, create a schedule, view the output, and create performance reports.

Creating a schedule consists of a number of steps. First, the current status of all of the lots in the test area is determined from the pre-shift lot status report. Lots of analog devices and lots in a stores operation are ignored.

Second, the lot status information is combined with the PROP file, which converts lots into jobs. The job consists of relevant lot information and a number of operations. The lot information includes product class, due date, and lot quantity. For each operation a processing time is computed along with a remaining cycle time.

Third, the user must check the current shop settings on resources, dispatching rules, and employee breaks. The user is shown the default settings and has the opportunity to make any changes.

Fourth, the schedule is created using the job, shop, and process information collected to this point. This schedule can be created using the genetic algorithm, which searches over combinations of dispatching rules, or the dispatching rules chosen by the user. In the latter case, no search is performed.

The genetic algorithm begins with a population of randomly created policies. The genetic operators shift the population toward more successful policies (measured on their ability to create a schedule with more on-time jobs). The algorithm stops after a fixed period of time; this time was selected after experimentation into the trade-off between schedule quality and search effort.

In addition to creating a shift schedule, the system creates a number of other files that are used to drive performance reports. There are three performance reports: shift summary, daily summary, and daily detailed. The summary reports compare what was done with what was scheduled. The scheduled operations for each machine are compared to the operations that have been performed on those lots. The detailed report compares the scheduled processing times to the actual times that are derived from daily lot history files.

If the user makes no changes to the default shop settings, the entire process can take as little as ten minutes. This compares to the fifteen minutes necessary in SIS to sequence just the lots waiting at one work center (the sort is slow due to the computing overhead associated with the massive SIS software). Therefore, the process can be done as part of the pre-shift planning, and the user has ample opportunity to update any data and to experiment with different shop settings.

4.6. *Implementation Issues*

During the development and implementation of the system, we had to continually compromise between the accuracy of the models and the requirements of the system. These compromises balanced the variation of processing times, the quest for perfection, the benefit of simplicity, the timeliness of data, and the use of a genetic algorithm for optimization.

Processing Times. One primary problem was a result of our attempt to model an uncertain process with a deterministic procedure. We wanted to use processing times to set reasonable targets for the test floor. Thus, we needed good estimates of what the processing times should be. The collection of these estimates was an important concern. We had historical data from the factory control system, which monitors when each lot begins and ends each operation, and developed estimation equations that use package type and lot size to calculate a processing time.

Perfection. We learned that perfection is an unattainable goal in a setting as complex as semiconductor test. However, we also learned that falling short of perfection is sufficient if our system improves the ability of the facility to satisfy customer demand efficiently. From the beginning, management knew (better than the project team did) that we could not hope to capture all of the activities that occur. This attitude allowed us to build a significant model instead of being overwhelmed by the complexity of finding an optimal solution. Since modeling and controlling the test area perfectly was beyond our reach, we concentrated on developing a system which meets the needs of the test area planners and provides them with a tool which they can use to intelligently manage their facility.

Simplicity. A significant feature of the implemented system is the ability of production planners to modify data related to resources and other factors. This called for a number of modules that could manipulate the data without requiring undue effort from the planners. These modules were user-friendly (easy to understand and fail-safe), flexible routines that the planners felt comfortable using. We feel that this has contributed to the success of the implementation.

Timeliness. Next to the estimates of processing times, accurate information about lot status was the hardest data to gather. Since there are hundreds of lots in the facility's inventory, determining which lots are waiting at which stations is not a trivial task. The scheduling system depends upon WIP extracts that are run from the factory control system before each shift. Although the planners have the ability through the factory control system to check on the status of individual lots, it was not possible to develop procedures to gather the status of the lots during the middle of the shift due to the amount of work the CIM system would have to perform.

Because of this deficiency, the scheduling system could react to unforeseen events only by rebuilding the original schedule from the beginning of the shift and incorporating the new information about machine breakdowns and lot arrivals. Searches to find good schedules were not possible. Accurate updated schedules could be found (with or without a search); however, if the lot status extract could be constructed quickly during the shift.

Optimization Procedure. A number of other researchers continue to work on ways to solve the job shop scheduling problem. Although there is no guarantee that the heuristic space genetic algorithm is the best procedure for finding a good shop schedule, a search over dispatching rules seemed to be an ideal approach for the needs and requirements of the test area.

The system makes use of the scheduling model and a separate genetic algorithm program. The two procedures are distinct modules developed and modified independently, yielding a great deal of flexibility during the testing and implementation of the scheduling system. We chose the genetic algorithm to gain the power of its parallelism and its simplicity and to avoid the premature convergence of local searches. The test area personnel could understand how policies combined to form new policies and how these policies could be used to create schedules.

4.8. *Contributions of Global Scheduling*

The test area now has a tool that can substantially improve scheduling through the combination of global information, a detailed scheduling model, a genetic search for good schedules, rescheduling after unforeseen events, and short-term performance reports.

According to the test area managers, the self-directed work teams using the scheduling system in the test area are maintaining a 90–96% on-time delivery rating; previously, they were using a manual scheduling system, and the test area had a 75–85% rating. Based on these results and other improvements, the company is expanding to other production facilities the effort to improve customer satisfaction through enhanced scheduling procedures.

The global system has a number of strengths compared with SIS, the previous scheduling system. As a centralized procedure, it can make use of information from around the test area including processing times, queue lengths, current setups, resource availability, and job arrivals. Testing on the shift schedules created showed that the genetic algorithm finds schedules that have more on-time lots than schedules created the fixed set of dispatching rules.

The schedule produced by the system gives the test area an overview of what should happen during the shift. After the shift ends, the schedule can be used to measure the performance of that shift. In fact, the ability to accurately model the test area and automatically compute a shift schedule was just as important to the test area as the ability to find better schedules. The effort spent on creating shift schedules has been reduced from 120 to 15 hours per week. The simulation component of the system allows the test area planners to forecast how modifying the level of available resources will affect the output of the test area. The system can also react to certain unforeseen events that may necessitate a change in the schedule.

5. Summary and Conclusions

This paper describes the development of a global scheduling system that uses a genetic algorithm to find good schedules. This system has been successfully implemented in a semiconductor test area. Controlling the test area is a complex, dynamic job shop scheduling problem where it is difficult to meet the management objectives of satisfying customer demand on-time and increasing throughput. The scheduling system uses the extensive data available in the CIM databases in order to simulate the operation of the test area. This system uses a genetic algorithm to search for combinations of dispatching rules that yield better schedules than those created by fixed dispatching rules. The primary accomplishments of this research are the implementation in a manufacturing environment of an advanced job shop scheduling system that uses modern information technology and the new heuristic procedure that searches the combinations of dispatching rules to find a better schedule.

This approach could be extended to any manufacturing area that has a complicated shop scheduling problem and a computer-integrated factory control system that can supply the necessary data for such a global scheduling system. In fact, the semiconductor manufacturing firm where we have implemented the system is considering exporting this system to other areas.

Additional research needs to focus on creating systems that manage uncertainty more explicitly, react confidently to unforeseen events, adapt readily to a changing manufacturing setting, and use more satisfactory processing time estimates. As a heuristic space procedure, the genetic algorithm could be used with other procedures that use control parameters to generate schedules.[1]

[1] This work was supported by a National Science Foundation Graduate Research Fellowship and by Harris Semiconductor and National Science Foundation grant DDM-9201627. The authors acknowledge the efforts of the personnel at Harris Semiconductor and the tireless work of the other project team members: Marijean Azrak, Tom McCutchen, and Joyce Townsend. Finally, the authors thank the editor and the referees for their useful comments.

References

ADAMS, J., E. BALAS, AND D. ZAWACK (1988), "The Shifting Bottleneck Procedure for Job Shop Scheduling," *Management Science,* 34, 391–401.

ADELSBERGER, H. H. AND J. J. KANET (1991), "The Leitstand—A New Tool in Computer-Integrated Manufacturing," *Production and Inventory Management,* 32, No. 1, 43–48.

ADLER, L., N. FRAIMAN, E. KOBACKER, M. PINEDO, J. C. PLOTNICOFF, AND T.-P. WU (1993), "BPSS: A Scheduling Support System for the Packaging Industry," *Operations Research,* 41, No. 4, 641–648.

BARNES, J. W. AND J. B. CHAMBERS (1991), *Solving the Job Shop Scheduling Problem Using Tabu Search,* Technical Report OPR91-06, Graduate Program in Operations Research, University of Texas at Austin.

BEAN, J. C. (1994), "Genetics and Random Keys for Sequencing and Optimization," *ORSA Journal of Computing,* 6, 154–160.

BENSANA, E., G. BEL, AND D. DUBOIS (1988), "OPAL: A Multi-Knowledge-Based System for Industrial Job-Shop Scheduling," *International Journal of Production Research,* 26, No. 5, 795–819.

BHASKARAN, K. AND M. PINEDO (1991), "Dispatching" in *Handbook of Industrial Engineering,* G. Salvendy, (ed.), J. Wiley, New York, Chapter 83.

CERNY, V. (1985), "Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm," *Journal of Optimization Theory and Applications,* 45, 41, 1985.

Consilium, Inc. (1988), *Short-Interval Scheduling System User's Manual,* Internal publication, Mountain View, CA.

DAVIS, L. (1985), "Job Shop Scheduling with Genetic Algorithms," in *Proceedings of an International Conference on Genetic Algorithms and their Applications,* J. Grefenstette, ed. Pittsburgh, Pennsylvania, pp. 136–140.

——— (ed.) (1991), *Handbook of Genetic Algorithms,* Van Nostrand Reinhold, New York.

FAALAND, B. AND T. SCHMITT (1993), "Cost-Based Scheduling of Workers and Equipment in a Fabrication and Assembly Shop," *Operations Research,* 41, No. 2, 253–268.

FISHER, H. AND G. L. THOMPSON (1963), "Probabilistic Learning Combinations of Local Job-Shop Scheduling Rules" in *Industrial Scheduling,* J. F. Muth and G. L. Thompson (eds.), Prentice-Hall, Inc., Englewood Cliffs, NJ. pp. 225–251.

FOX, B. R. AND M. B. MCMAHON (1990), "Genetic Operators for Sequencing Problems," Planning and Scheduling Group, McDonnell Douglas Space Systems, Houston, TX.

——— AND S. F. SMITH (1984), "ISIS—A Knowledge-Based System for Factory Scheduling," *Expert Systems,* 1, No. 1, 25–49.

FRY, T. D., P. R. PHILIPOOM, AND J. H. BLACKSTONE (1988), "A Simulation Study of Processing Time Dispatching Rules," *Journal of Operations Management,* 7, 77–92.

GLASSEY, C. R. AND R. G. PETRAKIAN (1989), *The Use of Bottleneck Starvation Avoidance With Queue Predictions in Shop Floor Control,* Research Report ESRC 89-23, University of California, Berkeley.

——— AND M. G. C. RESENDE (1988), "Closed-Loop Job Release Control for Visi Circuit Manufacturing," *IEEE Transactions on Semiconductor Manufacturing,* 1, 36–46.

GLOVER, F. (1977), "Heuristics for Integer Programming Using Surrogate Constraints," *Decision Sciences,* 8, No. 1, 156–166.

GOLDBERG, D. E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning,* Addison-Wesley, Reading, MA.

HACKMAN, S. T. AND R. C. LEACHMAN (1989), "A General Framework for Modeling Production," *Management Science,* 35, No. 4, 478–495.

HERRMANN, J. W. AND C.-Y. LEE (1992), *Three-Machine Look-Ahead Scheduling Problems,* Research Report 92-23, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL.

——— (1993), "Solving a Class Scheduling Problem With a Genetic Algorithm," to appear in *ORSA Journal of Computing.*

HOLLAND, J. H. (1975), *Adaptation in Natural and Artificial Systems,* University of Michigan Press, Ann Arbor, MI.

HUNG, Y.-F. AND R. C. LEACHMAN (1992), *A Production Planning Methodology for Semiconductor Manufacturing Based on Iterative Simulation and Linear Programming Calculations,* Report 92-26, Engineering Systems Research Center, University of California, Berkeley, CA.

JACOBS, F. R. (1984), "OPT Uncovered: Concepts Behind the System," *Industrial Engineering,* 16, No. 10, 32–41.

KIRKPATRICK, S., C. D. GELATT, AND M. P. VECCHI (1983), "Optimization by Simulated Annealing," *Science,* 220, 671.

VAN LAARHOVEN, P. J. M., E. H. L. AARTS, AND J. K. LENSTRA (1988), *Job Shop Scheduling by Simulated Annealing,* Report OS-R8809, Centre for Mathematics and Computer Science. Amsterdam, The Netherlands.

LEACHMAN, R. C. (1993), "Modeling Techniques for Automated Production Planning in the Semiconductor Industry" in *Optimization in Industry,* T. A. Ciriani and R. C. Leachman (eds.), John Wiley & Sons Ltd.

——— AND V. S. SOHONI, *Automated Shift Scheduling as a Tool for Problem Identification and People Management in Semiconductor Factories,* University of California, Berkeley, CA.

———, M. SOLORZANO, AND C. R. GLASSEY (1988), *A Queue Management Policy for the Release of Factory Work Orders,* Research Report 88-19, Engineering Systems Research Center, University of California at Berkeley.

LEE, C.-Y. AND J. W. HERRMANN (1993), *A Three-Machine Scheduling Problem With Look-Behind Characteristics,* Research Report 93-11, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL.

———, L. A. MARTIN-VEGA, R. UZSOY, AND J. HINCHMAN (1993), "Implementation of a Decision Support System for Scheduling Semiconductor Testing Operations," *Journal of Electronic Manufacturing,* 3, 121–131.

———, R. UZSOY, AND L. A. MARTIN-VEGA (1992), "Efficient Algorithms for Scheduling Semiconductor Burn-in Operations," *Operations Research,* 40, No. 4, 764–775.

LIEPINS, G. E. AND M. R. HILLIARD (1989), "Genetic Algorithms: Foundations and Applications," *Annals of Operations Research,* 21, 31–58.

MATSUO, H., C. J. SUH, AND R. S. SULLIVAN (1988), *Controlled Search Simulated Annealing for General Job Shop Scheduling Problem,* Working Paper #03-04-88. Graduate School of Business, University of Texas, Austin, Texas.

MONMA, C. L. AND C. N. POTTS (1989), "On the Complexity of Scheduling With Batch Setup Times," *Operations Research,* 37, No. 5, 798–804.

MORTON, T. E. (1993), *Heuristic Scheduling Systems,* Graduate School of Industrial Administration, Carnegie Mellon University, John Wiley, New York.

———, S. R. LAWRENCE, S. RAJAGOPOLAN, AND S. KEKRE (1988), "SCHED-STAR: A Price-Based Shop Scheduling Module," *Journal of Manufacturing and Operations Management,* 1, 131–181.

NAJMI, A. AND C. LOZINSKI, *Managing Factory Productivity Using Object-Oriented Simulation for Setting Shift Production Targets in VLSI Manufacturing,* University of California, Berkeley, CA.

NAKANO, R. AND T. YAMADA (1991), "Conventional Genetic Algorithms for Job Shop Problems" in *Proceedings of the Fourth International Conference on Genetic Algorithms,* Morgan Kaufmann Publishers, Inc. San Diego, California, pp. 474–479.

OW, P. S. AND S. F. SMITH (1988), "Viewing Scheduling as an Opportunistic Problem-Solving Process," *Annals of Operations Research,* 12, 85–108.

PANWALKER, S. S. AND W. ISKANDER (1977), "A Survey of Scheduling Rules," *Operations Research,* 25, 45–61.

SADEH, N. (1991), *MICRO-BOSS: A Micro-Opportunistic Factory Scheduler,* Center for Integrated Manufacturing Decision Systems, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

SMITH, S. F., M. S. FOX, AND P. S. OW (1986), "Constructing and Maintaining Detailed Production Plans: Investigations into the Development of Knowledge-Based Factory Scheduling Systems," *AI Magazine,* 7, No. 4, 45–61.

STORER, R. H., S. Y. D. WU, AND R. VACCARI (1990), *Local Search in Problem and Heuristic Space for Job Shop Scheduling,* Working Paper, Department of Industrial Engineering, Lehigh University, Allentown, PA.

——— (1992), "New Search Spaces for Sequencing Problems with Application to Job Shop Scheduling," *Management Science,* 38, No. 10, 1495–1509.

UZSOY, R., C.-Y. LEE, AND L. A. MARTIN-VEGA (1992a), "A Review of Production Planning and Scheduling Models in the Semiconductor Industry, Part I: System Characteristics, Performance Evaluation and Production Planning," *IIE Transactions,* 24, 47–61.

——— (1992b), "Scheduling Semiconductor Test Operations: Minimizing Maximum Lateness and Number of Tardy Jobs on a Single Machine," *Naval Research Logistics,* 39, 369–388.

——— (1993), "A Review of Production Planning and Scheduling Models in the Semiconductor Industry, part II: Shop Floor Control," *IIE Transactions,* 26, (1994), pp. 44–55.

——— AND J. HINCHMAN (1991a), *Scheduling Semiconductor Testing Operations: Optimization and Approximation,* Proceedings, Joint U.S.-German Conference on New Directions for Operations Research in Manufacturing, Gaithersburg, MD, July 30–31. pp. 179–199.

——— AND P. A. LEONARD (1991b), "Production Scheduling Algorithms for a Semiconductor Test Facility," *IEEE Transactions on Semiconductor Manufacturing.* 4, (1991), pp. 271–280.

VEPSALAINEN, A. P. J. AND T. E. MORTON (1988), "Improving Local Priority Rules with Global Lead-Time Estimates," *Journal of Manufacturing and Operations Management,* 1, 102–118.

WEIN, L. M. (1988), "Scheduling Semiconductor Wafer Fabrication," *IEEE Transactions on Semiconductor Manufacturing,* 1, 115–130.

WU, S. D., R. H. STORER, AND P.-C. CHANG (1993), "One-machine Rescheduling Heuristics with Efficiency and Stability as Criteria," *Computers and Operations Research,* 20, No. 1, 1–14.