Int J Adv Manuf Technol (2006) 27: 1163–1169
DOI 10.1007/s00170-004-2296-z

ORIGINAL ARTICLE

Amit Kumar Gupta · Appa Iyer Sivakumar

Job shop scheduling techniques in semiconductor manufacturing

Received: 25 March 2004 / Accepted: 15 June 2004 / Published online: 3 November 2004

Abstract This paper presents a brief review on job shop scheduling techniques in semiconductor manufacturing. The manufacturing environment in a semiconductor industry is considered a highly complex job shop, involving multiple types of work centers, large and changing varieties of products, sequence-dependent setup times, reentrant process flow, etc., in a dynamic scheduling environment. Due to the stubborn nature of the deterministic job shop scheduling problem itself, many of the solutions proposed are of hybrid construction cutting across the traditional disciplines. The problem has been investigated from a variety of perspectives resulting in several analytical techniques combining generic as well as problem-specific strategies. In this paper, we seek to provide a brief overview of the problem, the techniques used and the researchers involved in solving this problem.

Keywords Job shop · Review · Scheduling · Semiconductor manufacturing

1 Introduction

Scheduling concerns the allocation of limited resources to tasks over time. It is a decision making process that has as a goal the optimization of one or more objectives. The problem of scheduling in semiconductor manufacturing is considered a complex job shop scheduling problem. A vast amount of literature exists on job shop scheduling because of the continuous and improved research efforts in this field over the last five to six decades. The research focus on scheduling in semiconductor manufacturing has only come into existence in the last 15 years or so and has become a very important issue at present for the overall growth of the economy, since the semiconductor industry has seen a phase of rapid advancement during this period.

Since the scheduling problem is not amenable to any particular solution, the frontiers of research in this area are vast. This review does not hope to cover every approach because of the wide range of research. But we do wish to find the broadest currents of research that have been important for the development of this area and to present the nature of this work. The reader will be able to see the primary aspects of research in semiconductor scheduling and will gain some understanding into the present state of the research. Also, this paper does not focus on covering all papers published on semiconductor scheduling, but on the coverage of scheduling techniques used in semiconductor manufacturing.

Essentially, most approaches to the semiconductor manufacturing scheduling problem can be classified into four categories: heuristic rules, mathematical programming techniques, neighborhood search methods, and artificial intelligence techniques. According to historical developments, heuristic rules were initially studied, then mathematical programming tech-

Table 1. A list of scheduling techniques in semiconductor manufacturing

Scheduling techniques		
1.	Dispatching heuristics/priority rule	
2.	Mathematical programming techniques	(a) Branch and bound method (b) Lagrangian relaxation-based approaches (c) Filtered beam search (d) Decomposition methods (e) Queuing network models
3.	Neighbourhood search methods	(a) Tabu search (b) Simulated annealing (c) Genetic algorithm
4.	Artificial intelligence (AI) techniques	(a) Expert-/knowledge-based systems (b) Artificial neural networks (c) Fuzzy logic (d) Petri net based approaches

A.K. Gupta (✉) · A.I. Sivakumar
Center for Supply Chain Management (CSCM),
School of Mechanical and Production Engineering (MPE),
Nanyang Technological University (NTU),
Nanyang Avenue, 639798 Singapore, Singapore
E-mail: amitkg@pmail.ntu.edu.sg,
Tel.: +65-6790224,
Fax: +65-6791859

niques provided a new direction for determining the schedules. Most recently there has been considerable interest in using neighborhood search and artificial intelligence techniques for scheduling approaches. In this paper, a brief introduction and application is offered for each of the four areas mentioned above. Each area is further categorized in various scheduling techniques as presented in Table 1. The paper gives a brief review of each technique categorically and ends with a general conclusion on job shop scheduling techniques.

2 Dispatching heuristics/priority rules

Dispatching rules have been extensively applied to the scheduling problems in semiconductor manufacturing. They are procedures designed to provide good solutions to complex problems in real-time. The terms dispatching rules, scheduling rules, sequencing rules, or heuristics are often used synonymously [1]. In a machine shop, for example, whenever a machine has been freed, a dispatching rule inspects the waiting jobs and selects the job with the highest priority. Some of the prominent dispatching heuristics are first-in-first-out (FIFO), shortest processing time (SPT), largest processing time (LPT), shortest setup plus processing time (SSPT), shortest remaining processing time (SRPT), shortest slack time, earliest job due date (Job-EDD), earliest operation due date (Operation-EDD), cost over time (COVERT), apparent tardiness cost with setups (ATCS), time-in-system (TIS), work in next queue (WINQ), etc. Research in dispatching rules has been active for several decades with the development of many new rules. Here, only a brief review is being presented relevant to scheduling in semiconductor manufacturing.

Johri [2] presented a method used to dispatch jobs in one of AT&T's wafer fabrication clean rooms for the objective of minimizing idle time on important facilities. For each job a slack time is defined as the amount of time the lot can be delayed in queue before it is needed at the next important facility group in its route. The lot with the smallest slack time is processed first. Uzsoy et al. [3] presented a computational study of dispatching rules for semiconductor testing operations for a variety of due-date and cycle time related performance measures. The dispatching rules examined in this study are FIFO, SPT, SSPT, SRPT, Job-EDD, Operation-EDD, COVERT, and ATCS. The performance measures considered are mean of cycle time, average tardiness, number of tardy jobs and maximum lateness. Their results indicated that no one rule performs well for all performance measures. Holthaus and Rajendran [4] presented five new and efficient dispatching heuristics as an additive combination of the process time, the total work-content of jobs in the queue of the next operation of a job, the arrival time and the slack of a job. By an extensive simulation study, they also observed that no single rule is effective in minimizing all measures of performance. Dabbas et al. [5] proposed a modified dispatching approach that combines multiple dispatching criteria into a single rule with the objective of simultaneously optimizing multiple objectives. They also validated their approach using two different fab models at different levels of complexity in semiconductor manufacturing systems.

Heuristic rules have strong advantages in that these are easy to understand, easy to apply, and require relatively little computer time. The primary disadvantage is that these can not hope for an optimal solution even for a single objective in dynamic job shop.

3 Mathematical programming techniques

Mathematical programming has been applied extensively to job shop scheduling, formulating the problems using integer programming, mixed-integer programming and dynamic programming. Until recently, the use of these approaches has been limited because of the NP-hardness of the scheduling problems [6]. Now, the decomposition of the scheduling problems in a number of subproblems, new solution techniques, and the high computational power of modern computers have enabled these approaches to be used on larger problems. Still, difficulties in the formulation of material flow constraints as mathematical inequalities and the development of generalized solutions have limited the use of these approaches. The following section gives the brief description of the major identified solution approaches in this category.

3.1 Branch and bound method

Two popular solution techniques for integer programming problems are branch and bound and Lagrangian relaxation. Morton and Pentico [7] summarize the branch and bound method as: "The basic idea of branching is to conceptualize the problem as a decision tree. Each decision choice point–a node–corresponds to a partial solution. From each node, there grows a number of new branches, one for each possible decision. This branching process continues until leaf nodes that can not branch any further are reached. These leaf nodes are solutions to the scheduling problems."

The branch and bound method involves two fundamental steps [8]. One is calculating lower bounds on the optimal solution of the problem, which guides the branching process. Second is the branching process, which divides a problem into two mutually exclusive, exhaustive and smaller subproblems. These subproblems can be further subdivided. Each branch can be searched until a complete solution has been obtained, then subsequent branches are only followed until the cost associated with that bound exceeds the cost of the presently best-known solution. The fundamental issue in the branch and bound method is choosing appropriate lower bounds and branching efficiency.

One of the main drawbacks of all branch and bound methods is the lack of strong lower bounds in order to cut branches of the enumeration tree as early as possible [9]. Although efficient bounding and pruning procedures have been developed to speed up the search, this is still a very computational intensive procedure for solving large scheduling problems. Sung and Choung [10] used a branch and bound algorithm for the

dynamic scheduling of batch processing machines at a single burn-in oven in semiconductor manufacturing industry, with the objective measure of maximum completion time (Makespan) of all jobs. Since the burn-in operations are generally considered as the bottleneck process in the final testing step, minimizing Makespan on the burn-in oven is a highly important issue for productivity enhancement in the entire IC chip manufacturing process. They have exploited a branch and bound algorithm for analyzing a dynamic case with different job-release times.

3.2 Lagrangian relaxation-based approaches

In Lagrangian relaxation techniques, the integer programming problems are solved by omitting specific integer-valued constraints and adding the corresponding costs (due to these omissions and/or relaxations) to the objective function. As with branch and bound, Lagrangian relaxation is computationally expensive for large scheduling problems. The general idea behind the Lagrangian relaxation approach is to decompose the large multiple-job scheduling problem to smaller job level subproblems. This is possible by relaxing the capacity constraints through the use of Lagrangian multipliers. Beginning times for each job are then determined subject to the capacity multipliers and the job specific processing and precedence requirements.

Luh and Hoitomt [11] described the general methodology of using Lagrangian relaxation techniques in the scheduling of manufacturing systems by decomposing each of the scheduling problems into job- or operation-level subproblems. Chen et al. [12] modeled an IC test floor scenario as an integer programming formulation with the objective of minimizing both earliness and tardiness subject to resource constraints, precedence constraints, and processing time requirements. It is then solved by the Lagrangian relaxation approach, which relaxes the resource constraints and precedence constraints. Chen and Hsia [13, 14] also used the Lagrangian relaxation approach in scheduling the IC sort and test facilities with multiple resources and precedence constraints and compared the results with those obtained from other heuristic dispatching rules. They reported that heuristic rules do take much less time to find feasible schedules, however, the Lagrangian relaxation approach does generate better scheduling results within an acceptable computation time range.

Kaskavelis and Caramanis [15, 16] used efficient Lagrangian relaxation algorithms for the industry size job shop problems with more than 10,000 resource constraints in semiconductor testing facilities. They introduced two new features in the Lagrangian multiplier updating procedure. First, estimation of a surrogate dual cost function and second, adaptive step size. Sun et al. [17] used this method for single machine scheduling with sequence dependent setup times. Sequence-dependent setup times are formulated as capacity constraints and are then relaxed using Lagrangian multipliers. Yang and Chang [18] formulated a multiobjective model for IC sort and test based upon the current information at any given instant and found the Pareto optimal solution using the Lagrangian relaxation technique. They have simultaneously considered the objectives of cycle time and on-time delivery, in an additive structure with fixed weighting parameters, and thus used the Lagrangian relaxation method to get a subglobal optimal solution for the deterministic model based on the current information. Further, they have proposed the use of this technique for more than two objectives.

3.3 Filtered beam search

Filtered beam search is a derivative of the branch and bound method. It tries to eliminate branches in an intelligent way so that not all branches have to be examined. It thus requires less computer time in comparison to the branch and bound method, but can no longer guarantee an optimal solution as the latter does. With beam search, only the most promising nodes at any level are selected as nodes to branch from. The remaining nodes at that level are discarded permanently.

De and Lee [19] used the filtered beam search technique as a solution strategy in the development of a knowledge-based scheduling system for the scheduling of semiconductor testing operations. The testing operations were scheduled on both parallel discrete workstations and parallel batch workstations for a minimum Makespan objective. In the search tree of scheduling process, the amount of search needed to generate the solution is controlled by using a filter width and a beam width.

3.4 Decomposition methods

Decomposition methods attempt to develop solutions to complex scheduling problems by decomposing them into a number of smaller subproblems, which are more tractable and easier to understand. Solutions are developed for each of the subproblems individually and then reassembled to constitute a solution to the original problem [20]. The solutions may be exact or approximate depending on the nature of the problem addressed. The shifting bottleneck heuristic is one major work-center-based decomposition approach, in which the job shop scheduling problem is decomposed into single machine subproblems.

Uzsoy et al. [21] proposed the production scheduling algorithms for scheduling semiconductor test operations by decomposing the problem into a number of workstations and using a disjunctive graph representation. They developed algorithms for minimizing maximum lateness and minimizing number of tardy jobs. Perry and Uzsoy [22] described an approach to scheduling of semiconductor testing operations, which combines a decomposition approach for the static problem with an event-driven rescheduling approach to handling the dynamic events in the system. Ovacik [23] presented a work-center-based decomposition approach in his thesis, which is merely one example of a much broader class of decomposition methods. The decomposed work centers are scheduled in order of criticality until all of them have been scheduled and a feasible schedule achieved. A network representation of the scheduling problem is used to model the interactions between the work centers so as to allow the solutions to be integrated into a solution to the job shop problem. Demirkol et al. [24] developed extended dispatching procedures and more powerful decomposition methods

to develop improved schedules for complex job shops of the type encountered in semiconductor wafer fabs and testing facilities. Demirkol and Uzsoy [25] evaluated the performance of decomposition procedures for semiconductor testing facilities with respect to several different scheduling criteria/multiple objectives. Ovacik and Uzsoy [20] provided various decomposition approaches for factory scheduling problems, focusing on the performance of work-center-based decomposition methods for the complex shop scheduling problems in semiconductor manufacturing. They used a disjunctive graph representation to model the interactions between the subproblems. Further, they discussed the shifting bottleneck procedure in more detail and presented the computational experiments examining the performance of various version of this procedure in the several shop environments including semiconductor testing processes.

3.5 Queuing network model

A natural way to model many manufacturing systems is as a multi-class queuing network as introduced by Baskett et al. [26]. In such a network, units of workflow through the system along prescribed routes, requiring tasks to be performed on them by resources. These units of work contend with each other for the attention of the resources and queue up in front of the resources to await processing when the resource is engaged in processing other units of work. The units of flow are differentiated by "class". Members of a different class may have different processing requirements at the same resource, as well as different routes through the network.

Connors et al. [27] presented a novel method for "what's next" scheduling of semiconductor manufacturing lines based on a deterministic fluid network model, capable of generating dynamic schedules. Kumar and Kumar [28] provided an introduction to the application of queuing network models to the design and analysis of reentrant flow lines in semiconductor wafer fabs. Further, they discussed queuing- network-based analytical tools for both performance evaluation as well as dynamic control of such systems and presented two ways of establishing stability of scheduling policies: quadratic functionals and fluid models. Hopp et al. [29] developed an optimized queuing network (OQNet) capacity planning tool for supporting the design of semiconductor fabrication facilities that make use of queuing network approximations and an optimization route incorporating the common features of semiconductor environment such as batch processes, re-entrant flows, multiple product class, etc. With this tool, they addressed the problem of minimizing the facility cost required to meet specific volumes and cycle time targets.

This section attempted to show a little of what has been done in terms of mathematical programming techniques. The references are certainly not exhaustive, but these will provide good starting points for further inquiry into any of the more specific area. The advantages and disadvantages of these techniques are practically the exact opposite of those for heuristic approaches. These techniques can provide optimal solutions, but these are limited by the necessary enumeration and hence computer memory and CPU time. Even for the most effective reduction techniques, the scheduling problem simply becomes too large.

4 Neighborhood search methods

Neighborhood search methods provide good solutions and offer possibilities to be enhanced when combined with other heuristics. Conceptually, these techniques continue to add small changes (perturbations) and evaluate schedules until there is no improvement in the objective function. When this happens, the procedure is stopped with a high-quality solution. Popular techniques that belong to this family include the Tabu search, simulated annealing, and genetic algorithms. Each of these has its own perturbation method, stopping rules, and methods for avoiding local optimum.

A procedure based on local search, in contrast with a global search procedure, does not guarantee an optimal solution. It usually attempts to find a better solution than the current one within the neighborhood of the current one. Two schedules are neighbors if one can be obtained through a well-defined modification of the other [30]. At each iteration a local search procedure performs a search within the neighborhood and evaluates the various neighboring solutions. The procedure either accepts or rejects a candidate solution as the next schedule to move to based on a given acceptance-rejection criterion.

4.1 Tabu search

The basic idea of Tabu search [31, 32] is to explore the search space of all the feasible scheduling solutions by a sequence of moves. A move from one schedule to another schedule is made by evaluating all candidates and choosing the best available option. Some moves are classified as Tabu (i.e., they are forbidden) because they either trap the search at a local optimum, or they lead to cycling (repeating part of the search) and a Tabu list is built up from the history of moves used during the search. These Tabu moves force exploration of the search space until the old solution area (e.g., local optimum) is left behind.

Geiger et al. [33] developed efficient heuristics and a Tabu search procedure to generate high quality solutions in reasonable computational time for a production scheduling problem arising at the wet etching process in a semiconductor wafer fabrication facility. They formulated the scheduling problem as a flow shop sequencing model with the objective of minimizing Makespan.

4.2 Simulated annealing

Simulated annealing [34] is based on the analogy to the physical process of cooling and recrystalization of metals with low free energy. The current state of the thermodynamic system is analogous to the current scheduling solution, the energy equation for the thermodynamic system is analogous to the objective function, and the ground state is analogous to the global optimum. Application of this physical process to scheduling problems is similar to that of the Tabu search: instead of selecting the neigh-

bor with the best objective function value, simulated annealing randomizes the selection of the next initial solution. The better the objective value of a neighboring solution, the better chance it stands to be selected as the next starting solution [35].

Yim and Lee [36] presented a new method based on the use of simulated annealing for scheduling cluster tools in semiconductor fabrication with the scheduling objective of minimizing Makespan. A cluster tool consists of a group of single wafer chambers organized around a wafer transport device, or robot. Due to constraints imposed by multiple routes of each wafer type and machines with no buffer, it is difficult to find an optimal solution. Therefore simulated annealing is used to seek the near-optimal sequence and machine assignment of the operations.

4.3 Genetic algorithm

Genetic algorithms (GA) [37] are an optimization methodology based on a direct analogy to Darwinian natural selection and mutations in biological reproduction. In principle, genetic algorithms encode a parallel search through concept space, with each process attempting coarse-grain hill climbing. Instances of a concept correspond to individuals of a species. Induced changes and recombinations of these concepts are tested against an evaluation function to see which ones will survive into the next generation.

In genetic algorithms, the neighborhood concept is not based on a single schedule, but rather on a set of schedules. Hence, Tabu search and simulated annealing may be regarded as special cases of genetic algorithms with a population size equal to unity [30]. Thus, the genetic algorithms are more general and powerful techniques in neighborhood searches than others. Cavalieri et al. [38] presented an interesting application of genetic algorithms to solving a job shop scheduling problem in a semiconductor manufacturing system at ST Microelectronics plant. The algorithm guarantees the compatibility of the solution with online scheduling with a clearly better performance than that of the FIFO approach currently used in the plant. Lam et al. [39] developed a genetic algorithm for the problem of scheduling jobs to engineers with precedence relations and compatibility constraints in order to minimize the product design time (Makespan) in a semiconductor manufacturing company. Wang and Uzsoy [40] used a dynamic programming algorithm to develop a genetic algorithm for the problem of minimizing maximum lateness on a batch-processing machine in presence of dynamic job arrivals. They used a random keys encoding scheme in this problem as it eliminates the possibility of infeasible sequences with duplicate jobs, or jobs that are not scheduled at all, arising during the search. The problem they studied arises as a subproblem in work-center-based decomposition procedures for scheduling semiconductor manufacturing facilities.

These neighborhood search methods are gaining importance in solving the scheduling problems, since the computation time for the solution is not as significant a factor as in mathematical programming techniques. The possibility of local optima is one of the primary disadvantages of these methods.

5 Artificial intelligence techniques

Artificial Intelligence (AI) offers a means to combine various types of knowledge in a manner that can be used in the scheduling environment, i.e., use of human insights/intelligence imbedded into the system. In other words, the emphasis of AI methods is to solve a problem using methods that "appear" intelligent. AI techniques have four main advantages. First is the use of both qualitative and quantitative knowledge in the decision making process. Second is their capability to generate the heuristics that are more complex than the simple dispatching rules. The third is that the selection of the best heuristic can be based on information about the entire job shop. Fourth, they capture complex relationships in elegant new data structures and contain special techniques for powerful manipulation of the information in these data structures. However, the serious disadvantages include large time consumption, they are difficult to maintain and change, lack of ability to generate feasible solutions without checking the optimality, and they are system specific, i.e., there are no generic AI systems.

5.1 Expert/knowledge-based systems

Expert- and knowledge-based systems consist of two parts: a knowledge base and an inference engine to operate on that knowledge base. Formalizations of the "knowledge" that human experts use – into rules, procedures, heuristics, and other types of abstractions – are captured in the knowledge base. The inference engine selects a strategy to apply to the knowledge bases to solve the problem at hand. Yen and Pinedo [41] discussed a number of issues related to the design and development of scheduling systems for manufacturing. They discussed database and knowledge base issues, schedule generation issues, scheduling engines, various forms of user interfaces, and implementation issues for the scheduling systems.

De and Lee [19] proposed a knowledge-based scheduling system for the scheduling of semiconductor testing operations, considering it as a workstation scheduling problem that can be characterized as a general job shop scheduling problem with both parallel workstation clusters and batch processors. The proposed system consisted of two distinct components: a knowledge base developed using a frame-based knowledge representation scheme, and a solution strategy based on filtered beam search. Aytug et al. [42] presented a review of machine learning in scheduling and motivated the need for systems employing artificial intelligence methods for scheduling.

5.2 Artificial neural networks

Neural networks, also called distributed parallel processing models, have been studied in an attempt to mirror the learning and prediction abilities of human beings [43]. Neural network models are distinguished by network topology, node characteristics, and training or learning rules. The basic idea is the

recognition of patterns and rules that are embedded in a good schedule, and therefore have quick recognition of a good schedule. Artificial neural networks accomplish this task through the use of formally defined network topography and rules for "firing neurons". This pattern of firing neurons is observed on a set of "training" problems where the artificial neural network is presented with a problem and an acceptable solution. The trained network is then presented with a new problem, and based on the structure of the network and firing rules, it determines a solution as the "recommended" solution.

Neural networks are gaining importance in solving the job shop scheduling problems with several job types, exhibiting different arrival patterns, process plans, precedence sequences and batch sizes. Zhang and Huang [44] presented a survey of applications of neural networks in manufacturing, which also covers the subject of job shop scheduling. But unfortunately, neural networks have not yet been explored in scheduling the highly complex processes in semiconductor manufacturing environment to the best of our knowledge.

5.3 Fuzzy logic

Fuzzy set theory [45] can be useful in modeling and solving job shop scheduling problems with uncertain processing times, constraints, and setup times. These uncertainties can be represented by fuzzy numbers that are described by using the concept of an interval of confidence. These approaches are usually integrated with other methodologies (e.g., search procedures, constraint relaxation).

Kuroda and Wang [46] presented the potential value of applying fuzzy theory to real-life job shop scheduling problems using fuzzy information regarding due dates and/or operation times, technological constraints, etc. Azzaro-Pantel [47] used this fuzzy approach for performance modeling in a batch plant of semiconductor manufacturing. A discrete-event simulation model (MELISSA) for performance evaluation has been developed to treat uncertainties modeled by fuzzy numbers. Inclusion of fuzzy techniques provided the decision-maker with a range of possible values for completion times, average storage times, and operator workload instead of a unique value (which has little significance due to the variety of human operators).

5.4 Petri-net-based approach

A Petri net is defined as a bipartite directed graph containing places, transitions, and directed arcs connecting places to transitions and transitions to places. Places are used to represent conditions or the status of a component in a system, while the transitions represent the events or operations. At any given time, the distribution of tokens on places, called Petri net marking, defines the current state of the modeled system. A marked Petri net can be used to study the dynamic behavior of the modeled discrete event system that exhibit sequential, concurrent, and conflicting relations among the events and operations.

Zhou and Jeng [48] presented a tutorial on Petri nets, introducing definitions and concepts of Petri nets, discussing system modeling, presenting their properties and analysis methods, and reviewing applications of Petri nets in semiconductor manufacturing automation. They also presented a case study to show how a Petri net methodology can be applied to a complex real world system, by illustrating through Petri net modeling, analysis, and simulation, of the photolithography area of an IC wafer fab. Xiong and Zhou [49] proposed two Petri net-based hybrid heuristic search strategies and their application to semiconductor test facilities scheduling with a size limit of 79 resources and 30 jobs. Odrey et al. [50] presented a modeling approach for a reentrant flow manufacturing line, as common in semiconductor wafer fabs and test facilities, based on the use of generalized Petri nets.

These artificial intelligence techniques can provide optimal solutions in short computation times as compared to heuristic approaches and mathematical programming techniques. The primary disadvantage with these techniques is getting trapped in the local optima. Another problem is of lack of insight into the nature of the solution in using these techniques.

6 Concluding remarks

This paper presented a brief review of the scheduling techniques in scheduling the semiconductor manufacturing processes. These include dispatching heuristics, mathematical programming techniques, neighborhood search methods, and AI techniques, for the purpose of broad categorization. Dispatching heuristics provide schedules quickly, but there is no guarantee of optimality using these rules. Mathematical programming techniques can provide optimal solutions, but computation time is prohibitive for large problem sets. Neighborhood search methods are capable of finding near-optimal solutions in a reasonable amount of computation time, but these have the disadvantage of getting trapped in local optima. AI techniques also suffer, with the problem of being trapped at local optima without proper insight into the nature of the solution. This paper has summarized the wide spectrum of major techniques used to solve the scheduling problems in semiconductor manufacturing processes. Although the references are certainly not exhaustive, these will provide good starting points for further inquiry into any of the more specific areas.

Acknowledgement The authors gratefully acknowledge Singapore Millennium Foundation (SMF) and Nanyang Technological University (NTU), Singapore, for providing the necessary funding and required infrastructure for carrying out this research.

References

1. Panwalker S, Iskander W (1997) A survey of scheduling rules. Oper Res 25(1):45–61
2. Johri PK (1989) Dispatching in an integrated circuit wafer fabrication line. In: MacNair EA, Musselman KJ, Heidelberger P (eds) Proceedings of the 1989 Winter Simulation Conference, pp 918–921
3. Uzsoy R, Church LK, Ovacik IM (1992) Dispatching rules for semiconductor resting operations), a computational study. IEEE/CHMT International Electronics Manufacturing Technology Symposium, IEEE, pp 272–276

4. Holthaus O, Rajendran C (1997) Efficient dispatching rules for scheduling in a job shop. Int J Prod Econ 48:87–105
5. Dabbas RM, Chen H-N, Fowler JW, Shunk D (2001) A combined dispatching criteria approach to scheduling semiconductor manufacturing systems. Comput Ind Eng 39:307–324
6. Baker KR (1974) Introduction to sequencing and scheduling. Wiley, New York
7. Morton TE, Pentico DW (1993) Heuristic scheduling systems. Wiley, New York
8. Sellers DW (1996) A survey of approaches to the job shop scheduling problem. Proceedings of the Twenty-Eighth Southeastern IEEE Symposium on System Theory, pp 396–400
9. Blazewicz J, Domschke W, Pesch E: The job shop scheduling problem: conventional and new solution techniques. Eur J Oper Res 93: 1–33
10. Sung CS, Choung YI (2000) Minimizing makespan on a single burn-in oven in semiconductor manufacturing. Eur J Oper Res 120:559-574
11. Luh PB, Hoitomt DJ (1993) Scheduling of manufacturing systems using the Lagrangian relaxation technique. IEEE Trans Semicond Manuf 38(7):1066-1079
12. Chen T-R, Chen C-W, Kao J (1993) Due windows scheduling for ic sort and test facilities with precedence constrains via Lagrangian relaxation. IEEE/SEMI International Semiconductor Manufacturing Science Symposium, pp 110–114
13. Chen T-R, Hsia TC (1994) Job shop scheduling with multiple resources and an application to a semiconductor testing facility. Proceedings of the 33rd IEEE Conference on Decision and Control 2:1564–1570
14. Chen T-R, Hsia TC (1997) Scheduling for IC sort and test facilities with precedence constrains via Lagrangian relaxation. J Manuf Syst 16(2):117–128
15. Kaskavelis CA, Caramanis MC (1994) Application of a Lagrangian relaxation based scheduling algorithm to a semiconductor testing facility. Proceedings of the Fourth International Conference on Computer Integrated Manufacturing and Automation Technology, IEEE, pp 106–112
16. Kaskavelis CA, Caramanis MC (1998) Efficient Lagrangian relaxation algorithms for industrial size job-shop scheduling problems. IIE Trans 30:1085–1097
17. Sun X, James SN, Klein CM (1999) Single-machine scheduling with sequence dependent setup to minimize total weighted squared tardiness. IIE Trans 31:113–124
18. Yang J, Chang T-S (1998) Multiobjective scheduling for ic sort and test with a simulation testbed. IEEE Trans Semicond Manuf 11(2): 304–315
19. De S, Lee A (1998) Towards a knowledge-based scheduling system for semiconductor testing. Int J Prod Res 36(4):1045–1073
20. Ovacik IM, Uzsoy R (1997) Decomposition methods for complex factory scheduling problems. Kluwer, Boston
21. Uzsoy R, Martin-Vega LA, Lee C-Y, Leonard PA (1991) Production scheduling algorithms for a semiconductor test facility. IEEE Trans Semicond Manuf 4(4):270–280
22. Perry CN, Uzsoy R (1993) Reactive scheduling of a semiconductor testing facility. IEEE/CPMT International Electronics Manufacturing Technology Symposium, pp 191–194
23. Ovacik IM (1994) A decomposition methodology for scheduling complex job shops. Dissertation, Purdue University
24. Demirkol E, Uzsoy R, Ovacik IM (1995) Decomposition algorithms for scheduling semiconductor testing facilities. IEEE/CPMT International Electronics Manufacturing Technology Symposium, pp 199–204
25. Demirkol E, Uzsoy R (1997) Performance of decomposition methods for complex workshops under multiple criteria. Comput Ind Eng 33(1-2):261–264
26. Baskett F, Chandy KM, Muntz RR, Pala-Cios FG (1975) Open, closed, and mixed networks of queues with different classes of customers. J Assoc Comput Mach 22(2):248–260
27. Connors D, Feigin G, Yao D (1994) Scheduling semiconductor lines using a fluid network model. IIE Trans Robot Autom 10(2):88–98
28. Kumar S, Kumar PR (2001) Queuing network models in the design and analysis of semiconductor wafer fabs. IIE Trans Robot Autom 17(5):548–561
29. Hopp WJ, Spearman ML, Chayet S, Donohue KL, Gel ES (2002) Using an optimized queuing network model to support wafer fab design. IIE Trans 34:119–130
30. Pinedo M (1995) Scheduling, theory, algorithms and systems Prentice-Hall, Englewood Cliffs, NJ
31. Glover F (1989) Tabu search – part I. ORSA J Comput 1(3):190–206
32. Glover F (1990) Tabu search – part II. ORSA J Comput 2(1):4–32
33. Geiger CD, Kempf KG, Uzsoy R (1997) A Tabu search approach to scheduling an automated wet etch station. J Manuf Syst 16(2):102–116
34. Kirkpatrick S, Gelatt CD Jr, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680
35. Kiran AS (1998) Simulation and scheduling. In: Banks J (ed) Handbook of simulation. Wiley, New York, pp 677–717
36. Yim SJ, Lee DY (1999) Scheduling cluster tools in wafer fabrication using candidate list and simulated annealing. J Intell Manuf 10(6):531–540
37. Goldberg D (1988) Genetic algorithms in search optimization and machine learning. Addison-Wesley, Menlo Park, CA
38. Cavalieri S, Crisafulli F, Mirabella O (1999) A genetic algorithm for job-shop scheduling in a semiconductor manufacturing system. IEEE Trans Eng Manage 41(2):957–961
39. Lam FSC, Lin BC, Sriskandarajah C, Yan H (1999) Scheduling to minimize product design time using a genetic algorithm. Int J Prod Res 37(6):1369–1386
40. Wang CS, Uzsoy R (2002) A genetic algorithm to minimize maximum lateness on a batch processing machine. Comput Oper Res 29: 1621–1640
41. Yen BP-C, Pinedo M (1994) On the design and development of scheduling systems. Proceedings of the Fourth International Conference on Computer Integrated Manufacturing and Automation Technology, IEEE, pp 197–204
42. Aytug H, Bhattacharyya S, Koehler GJ, Snowdon JL (1994) A review of machine learning in scheduling, IEEE Trans Eng Manage 41(2):165–171
43. Zurada JM (1992) Introduction to artificial neural systems West Publishing, St. Paul, MN
44. Zhang HC, Huang SH (1995) Applications of neural networks in manufacturing: a state-of-the-art survey. Int J Prod Res 33(3):705–728
45. Zadeh LA (1965) Fuzzy sets. Inf Control 8:338–353
46. Kuroda M, Wang Z (1996) Fuzzy job shop scheduling. Int J Prod Econ 44:45–51
47. Azzaro-Pantel C, Floquet P, Pibouleau L, Domenech S (1997) A fuzzy approach for performance modeling in a batch plant: application to semiconductor manufacturing. IEEE Trans Fuzzy Syst 5(3):338–357
48. Zhou MC, Jeng MD (1998) Modeling, analysis, simulation, scheduling, and control of semiconductor manufacturing systems: a Petri net approach. IEEE Trans Semicond Manuf 11(3):333–357
49. Xiong HH, Zhou MC (1998) Scheduling of semiconductor test facility via petri nets and hybrid heuristic search. IEEE Trans Semicond Manuf 11(3):384–393
50. Odrey NG, Green JD, Appello A (2001) A generalized Petri net modeling approach for the control of re-entrant flow semiconductor wafer fabrication. Robot Comput Integr Manuf 17:5–11