Proceedings of the 2000 Winter Simulation Conference
J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, eds.

MAXIMIZING DELIVERY PERFORMANCE IN SEMICONDUCTOR WAFER FABRICATION FACILITIES

Scott J. Mason

Department of Industrial Engineering
University of Arkansas
4207 Bell Engineering Center
Fayetteville, AR 72701, U.S.A.

John W. Fowler

Department of Industrial Engineering
Arizona State University
P. O. Box 875906
Tempe, AZ 85287-5906, U.S.A.

ABSTRACT

This paper is motivated by the problem of scheduling customer orders (jobs) in a semiconductor fabrication facility ("wafer fab") to maximize delivery performance when the jobs have non-identical priorities (weights). As each job is typically assigned a weight based on its size, value, and/or requesting customer, a wafer fab's delivery performance can be evaluated in terms of minimizing the sum of each job's weighted tardiness. A heuristic has been proposed for obtaining "good" solutions to this complex problem. Using a "real world" wafer fab data model, the heuristic is compared to a number of dispatching rules in terms of how well each method produces job sequences that maximize delivery performance of customer orders. Results suggest that the heuristic consistently produces the best overall schedules, but there is a price to be paid in terms of solution speed.

1 INTRODUCTION

In recent years, the number of applications and society's demand for integrated circuits (ICs) have increased dramatically. Microprocessors, memory chips, microcontrollers, and other semiconductor devices were once only the domain of corporate and research institutions; today, these devices are a part of everyone's lives, from personal computers, to cellular phones, to the projection system inside television sets. This increased demand has in turn caused high-tech microelectronics factories ("wafer fabs") to renew their efforts to provide high-quality deliveries to their customers. This paper is motivated by the problem of scheduling customer orders (jobs) in a wafer fab to maximize delivery performance when the jobs have non-identical priorities (weights).

Let L_j denote the lateness of job j, which is equal to the difference between the job's completion time C_j and its due date d_j. The tardiness of job j, T_j, equals $\max(L_j,0)$. Computing $\sum_j T_j$ provides an estimate of the wafer fab's delivery performance. However, all jobs typically are not considered equal in the wafer fab, as certain customers' orders are of greater importance or priority than others. Each job's weight is often a function of one or more of the following: size, value, and requesting or end customer. When more than a single job weight is present in the fab, each job's tardiness can be scaled by the job's corresponding priority or weight (w_j), then summed to evaluate the fab's delivery performance in terms of total weighted tardiness ($\sum w_j T_j$ or TWT).

Mason (2000) describes wafer fabs as "complex" job shops, as they contain job-specific reentrant flow across a number of unique tool groups (multiple, identical machines operating in parallel), some of which process jobs in batches, while others require sequence-dependent setups. The existence of individual lot (job) release or ready times and due dates combine with the other processing environment characteristics mentioned above to present a "complex" job shop scheduling problem.

Mason, Carlyle, and Fowler (2000) present a mixed integer program (MIP) heuristic for minimizing total weighted tardiness in wafer fabs, but found that their MIP heuristic's TWT solution degrades significantly as the size of the problem instances grows. Using a representative "mini-fab" model from the literature that succinctly cap tures the processing environment in a wafer fab (El Adl, Rodriguez, and Tsakalis 1996), the authors' MIP heuristic only produces feasible schedules for problem instances smaller than eight jobs. In an effort to obtain "good"-solutions to "real-world" wafer fab scheduling problems, Mason, Fowler, and Carlyle (2000) develop a heuristic that is capable of accommodating the processing complexities inherent in wafer fabs (see Section Two). The heuristic shows potential for maximizing a wafer fab's delivery performance, as it consistently produces schedules with

lower TWT than standard first-in/first-out processing for the mini-fab model. This paper presents the results from a case study in which "real world" wafer fab data is used to compare the heuristic to a number of dispatching rules in terms of how well each method produces job sequences that maximize delivery performance of customer orders

The remaining sections of this paper are organized as follows. Section Two presents an overview of the semiconductor manufacturing process, followed by a description of the proposed heuristic in Section Three. Section Four describes the case study's experimental design and results. Finally, Section Five presents conclusions and directions for future research.

2 SEMICONDUCTOR MANUFACTURING OVERVIEW

The process of manufacturing integrated circuits (semiconductors) on silicon wafer substrates is arguably one of the most expensive, complex manufacturing processes in existence. Uzsoy, Lee, and Martin-Vega (1992) and Kumar (1994) both provide an excellent description of the semiconductor manufacturing process. The semiconductor manufacturing process is performed in a clean environment known as a wafer fabrication facility or wafer fab. The steps required to manufacture a semiconductor product or device are described in a process flow or process routing; current generation semiconductor process flows can contain between 250 and 500 manufacturing or processing steps. Typically, the individual silicon wafers upon which semiconductors are manufactured are grouped into "lots" of 25 wafers. Each lot is uniquely identified in the wafer fab's manufacturing execution system (MES) as a unit of production (job). Therefore, the individual wafers within a lot travel together throughout the manufacturing process.

The equipment set used to manufacture semiconductors is typically made up of 60 to 80 different equipment types. These equipment types contain a diverse array of wafer processing tools in terms of the quantity of wafers that can be processed concurrently. While single wafer tools (photolithography steppers) only process one wafer at a time, other tools (acid bath wet sinks) can process entire lots concurrently. Wafer fabs also contain batching tools (diffusion furnaces) that can process multiple lots of wafers simultaneously. Finally, some tools (ion implanters) are subject to sequence-dependent setup times, as the time required to setup the tool depends upon the previous lot that was processed on the tool.

Due to the production volumes required in today's wafer fabs, these fabs contain multiple tools of each equipment type. This redundancy leads to the notion of a workstation or "tool group" made up of similar tools of a given equipment type that process wafers in parallel. Finally, semiconductor process flows contain a considerable amount of reentrant or recirculating flow, wherein a given tool may be visited a number of times during the manufacturing process by the same lot (job). This type of flow is necessitated by the capital cost of wafer processing equipment, which can cost up to $7,000,000 per tool.

3 A HEURISTIC FOR MAXIMIZING DELIVERY PERFORMANCE IN WAFER FABS

While numerous dispatching rules exist for sequencing jobs on single machines or tool groups, recent research efforts have focused on scheduling entire factories using decomposition methods. These methods, such as Adams, Balas, and Zawack's (1988) Shifting Bottleneck Procedure (SB), typically employ a "divide and conquer" approach by decomposing factory scheduling problems into smaller, more tractable tool group subproblems (Pinedo and Singer 1999). The most "critical" or bottleneck tool group is scheduled at each iteration.

Our modified Shifting Bottleneck heuristic (SBH) for maximizing delivery performance in a wafer fab is described as follows:

1. Let M denote the set of all m tool groups. Initially, the set of tool groups that have been sequenced or scheduled, M_0, is empty.
2. Identify and solve the subproblems for each tool group $i \in M \setminus M_0$.
3. Identify a critical or bottleneck tool group $k \in M \setminus M_0$.
4. Sequence tool group k using the subproblem solution from Step 2. Set $M_0 = M_0 \bigcup \{k\}$.
5. Optional: Reoptimize the schedule for each tool group $m \in M_0$, considering the newly added disjunctive arcs for tool group k.
6. If $M = M_0$, stop. Otherwise, go to Step 2.

Like the original SB procedure, our heuristic decomposes the wafer fab scheduling problem into smaller, more tractable subproblems. Depending upon the type of tool group being evaluated (single or multiple machines operating in parallel, a subset of which contain sequence-dependent setups, another subset of which are capable of batch processing), a different technique is applied during the solution of the tool group subproblems. Finally, as is the case with the original SB procedure, the re-optimization step in Step 5 is optional.

A mixed-level factorial experiment was performed to "tune" the performance of the heuristic in terms of producing low TWT schedules for wafer fab problems. Results indicate that scheduling batching tool groups before any other tool groups typically resulted in the

heuristic producing lower TWT solutions for the problem instances under consideration. This suggests the batching tool groups play a significant part in determining the TWT of each job in the complex job shop. Therefore, forming the "best" batches, regardless of the resulting idle time that may be inserted into each tool group's schedule, tends to produce schedules with low TWT. Therefore, this approach will be used in the following case study.

4 A CASE STUDY

As Mason, Fowler, and Carlyle's (2000) SBH consistently outperformed the same authors' MIP heuristic for the mini-fab model, subsequent experiments were run to test the heuristic's performance in scheduling a larger, more complex wafer fab. The solution quality and execution speed of the heuristic was compared with five well known dispatching rules in an attempt to measure each method's effectiveness in minimizing TWT (maximizing delivery performance) in a wafer fab:

- Priority-based First-In/First-Out (PRFIFO), which selects the job with the highest priority. Within equal priorities, the job that entered the tool group's queue first is favored.
- Least Slack (LSLACK), which schedules the job with the smallest slack, where slack is defined as the difference between the lot's due date and remaining processing time
- Priority-based Critical Ratio (PCR), which first schedules the lot with highest priority, then schedules the job with the lowest critical ratio. If the job's due date is greater than the current time, the critical ratio is computed as $\frac{1 + DueDate - Tnow}{1 + LTF * RPT}$, where $Tnow$ is the current time, LTF is the lead time factor and RPT is the job's remaining process time. Otherwise, the critical ratio is computed as $((1 + Tnow - DueDate * (1 + LTF * RPT))^{-1}$ (Wright Williams & Kelly 2000).
- Priority-based Earliest Due Date (PREDD), which selects the job with the highest priority. Within equal priorities, the job with the earliest due date is favored.
- Shortest Processing Time (SPT), which selects the job with the shortest process time.

4.1 SEMATECH Testbed Datasets

In the early 1990s, a group of researchers at SEMATECH, the United States' semiconductor research consortium, recognized that no representative wafer fab data existed in the public domain that could be used for testing new simulation packages, proposed heuristics and factory control policies, and other newly developed approaches to wafer fab/equipment scheduling. As part of SEMATECH's Measurement and Improvement of Manufacturing Capacity project, Fowler, Feigin, and Leachman (1995) developed a structured set of file formats for specifying wafer fab data. While not containing any distributional information, the testbed files provided a standardized way of capturing the process flow, rework, tool set, operator, and production volume requirements of a given wafer fab. Currently, there are seven factory datasets available at <http://www.eas.asu.edu/~masmlab>, along with a document detailing the format of each input file contained in the testbed.

Testbed Dataset 1 was used to assess the effectiveness of the heuristic and the five dispatching rules in minimizing TWT in a complex job shop. This dataset is composed of two different product flows and 83 different tool groups. Product 1 has 210 processing steps, while Product 2's process routing contains 245 steps. Preliminary analysis of Dataset 1 was conducted using Wright Williams & Kelly's Factory Explorer® v2.7 (2000) to identify the underlying factory's bottleneck tools, as scheduling efforts are typically focused on these tools.

Various modifications were made to the dataset to ensure the most critical or bottleneck tools were included in our investigation of a smaller subset of this production volume. First, the maximum batch size of each batching tool group was reduced from it's specified value to two lots. Next, the product mix in Dataset 1 was modified so that both products were released into the factory in equal amounts in an effort to promote a greater probability of tool group setups and better batching efficiencies. All tool group interruptions, wafer scrap and rework, and operators were removed from Dataset 1, as the SBH currently does not take any of these factors into account during its analysis.

Finally, in order to create a more realistic set of data for our analysis, the sequence-independent setup times associated with the medium- and high-current implanters in Dataset 1 were replaced with sequence-dependent setup times (Table 1). Specifically, if there is any change in the tool's required dopant species, 30 minutes of setup time is required. An additional 30 minutes is required to change from phosphorus (P) or arsenic (As) to boron (B) or boron difluoride (BF_2). Finally, changing the wafer's tilt angle requires 15 minutes. BF_2 and As steps have a tilt angle of 0 degrees, while B and P steps required 7 degrees of tilt.

Table 1: Ion Implanter Setup Matrix

Current Species	**Proposed Species**			
	P	As	BF_2	B
P	-	45	75	60
As	45	-	60	75
BF_2	45	30	-	45
B	30	45	45	-

4.2 Experimental Design

Once the factory's product mix, batching tool groups' maximum batch size, and sequence-dependent setups were modified, Factory Explorer's Capacity Analysis module was used to identify the top bottleneck tools of the modified dataset. The dataset's top 10 bottleneck tools with purchase prices exceeding $100,000 were selected for the comparative analysis. If a tool was identified as a constraint in the fab that cost less than this amount, the fab director would surely be able to obtain the funds necessary to purchase one or more additional copies of this tool type. Wafer fabs contain a number of tools that cost in excess of $1,000,000—these more expensive tools would never consistently be starved of work because of the existence of a sub-$100,000 constraint tool.

The resulting bottleneck tool list from modified Dataset 1 is given in Table 2 in descending order of constraint tool groups (30_DRIVE_OX is the modified dataset's bottleneck tool). The columns in the table list each tool group's name, tool type, and the number of tools to be used in the comparative analysis. A "serial" tool is one that processes one job at a time, while a "batch" tool is capable of processing multiple jobs simultaneously. Finally, "SDS" refers to tools that are subject to sequence-dependent setups.

Table 2: Modified Testbed Dataset 1's Bottleneck Tool List

Tool Group	Tool Type	Tool Qty
30_DRIVE_OX	Batch	5
67_MATRIX	Serial	5
76_E_SINK	Serial	3
10_MED_CURRENT_IMP	Serial (SDS)	2
11_HIGH_CURRENT_IMP	Serial (SDS)	2
14_PEAK	Serial	1
6_NONCRIT_DEV	Serial	5
5_CRIT_DEV	Serial	6
31_OXIDE_1	Batch	3
37_POLY_DEP	Batch	2

As only a percentage of Dataset 1's production volume will be used in the comparative analysis, the tool set quantities need to be reduced such that critical or bottleneck tools still existed in our experimental test cases. Preliminary experiments were conducted to determine the proper number of machines in each tool group for our analysis. The quantity of tools in tool group k was determined using the daily going rate (DGR) as calculated by Factory Explorer of a single tool in k. A tool's DGR is defined as the sum of product throughput rates divided by the tool's capacity loading (Wright Williams & Kelly 2000). In other words, DGR expresses the number production units or jobs that a tool can process in a day. The quantity of tools in tool group k was set equal to $DGR_{\max} / DGR_k$, where $DGR_{\max}$ is the maximum tool DGR across all ten tool groups and DGR_k is the DGR for a tool in group k. This specification of tool quantities was used to create a "balanced" tool set for use in the comparative analysis. Any fractional tool quantities were round up or down to the appropriate whole number of tools.

Finally, as only the top 10 bottleneck tool groups will be considered in our comparative analysis, all manufacturing steps that do not require processing on one of the tool groups listed in Table 2 will be treated as simple processing delays. Therefore, the tool required for processing each non-critical step is replaced with a generic "Delay" tool that has theoretically infinite capacity. This will shift the focus of the SBH and the previously mentioned dispatching rules to the factory's true bottleneck tool groups. Following this reduction procedure, Product 1's 210 processing steps are reduced to 73 steps, while 97 processing steps will now be used to represent Product 2's original 245 steps.

Ten test cases will be generated for a 20-job instance of the modified Dataset 1 model. In each case, the jobs will be divided evenly between Products 1 and 2. In each of the five different problem instances created, job j will have its weight w_j is uniformly distributed [1, 10], while its ready time r_j will have a 50% probability of being equal to zero and a 50% probability of being distributed according to [1, 300]. Finally, each job's due date d_j will be uniformly distributed in the interval (RawPT – 500, RawPT + 500), where RawPT denotes each product's raw process time (18,959 for Product 1 and 21,694 for Product 2). Each problem instance will be scheduled using the five dispatching rules discussed above and the proposed heuristic. The TWT of each of the six schedules will be measured to assess the ability of each methodology to minimize TWT in a complex job shop.

4.3 Experimental Results

Let $TWT(H,I)$ be the TWT value obtained by heuristic H on a problem instance I. Further, let $BEST(I) = \min_H [TWT(H,I)]$. Table 3 presents the results for the 20-job problem instances of the modified Dataset 1 experiments in terms of the ratio of $TWT(H,I)$ to $BEST(I)$. The average and variance of the $TWT(H,I)$ values are also displayed for each heuristic H.

As Table 3 indicates, the proposed heuristic consistently produces schedules with low TWT for the 20-job instance of modified Dataset 1. The SBH method has the lowest average ratio of $TWT(H,I)$ to $BEST(I)$, as

well as the minimum variance and median. This method produced the best schedule in nine of the ten test cases investigated. However, the heuristic's improved solution performance does not come without a price. While the dispatching rules investigated required five seconds to completely schedule all 20 jobs, SBH required an average of 4.5 minutes to produce its schedules.

Table 3: Ratio of $TWT(H,I)$ to $BEST(I)$ for the 20-Job Instance of Modified Dataset 1. $BEST(I) = \min_{H}[TWT(H,I)]$ is bolded.

Case	PRFIFO	LSLACK	PCR	PREDD	SPT	SBH
1	1.94	2.61	1.92	1.94	2.50	**1.00**
2	2.29	2.62	2.25	2.27	3.40	**1.00**
3	**1.00**	1.16	**1.00**	**1.00**	1.16	2.68
4	1.52	1.97	1.24	1.37	2.54	**1.00**
5	1.37	1.88	1.37	1.37	1.79	**1.00**
6	2.57	3.25	2.59	2.63	2.87	**1.00**
7	1.60	1.95	1.59	1.60	1.77	**1.00**
8	1.67	1.92	1.67	1.67	1.85	**1.00**
9	2.44	3.01	2.44	2.43	2.64	**1.00**
10	2.10	2.39	2.09	2.10	2.37	**1.00**
Avg	1.75	2.21	1.71	1.74	2.29	1.24
Var	0.30	0.46	0.33	0.33	0.58	0.40

5 CONCLUSIONS

The investigation into using the heuristic for scheduling larger wafer fabs suggest that while it consistently produces the best overall schedules in terms of TWT, a price must be paid in terms of solution speed. While the five dispatching rules only required five seconds to produce a schedule, the heuristic required 4.5 minutes to schedule 20 jobs. The SBH could be embedded into a wafer fab's manufacturing execution system (MES) to be run at the beginning of each workday or production shift (scheduling horizon). The heuristic would produce a proposed job sequence for each tool group in the factory, taking into account the expected movement of jobs within the factory over the scheduling horizon. However, future research is needed to speed up the proposed heuristic before it can be validated for use in scheduling an entire wafer fab.

Another area for investigation is to understand whether or not every tool group's subproblem needs to be solved explicitly. It may be possible to model non-critical processing tool groups, such as microscopes and spin-rinse dryers, as a simple processing delay, thereby reducing the number of tool group subproblems that must be evaluated. Initial experimentation using generic processing delays did not uncover any problems with taking this approach, although the variability inherent in a wafer fab is often under-represented when using any type of simulated processing delay approach. Care would need to be taken to specify processing delays that closely match the actual delays experienced in the wafer fab.

Another possible method for improving the heuristic's execution speed would be to recalculate the due date for only those nodes that are affected by the newly added disjunctive arcs, rather than for all nodes in the disjunctive graph. This should reduce the amount of time spent recalculating due dates at each iteration, and therefore, the time required to completely analyze the problem under consideration.

REFERENCES

Adams, J., E. Balas, D. Zawack. 1988. The shifting bottleneck procedure for job shop scheduling. *Management Science* 34:391-401.

El Adl, M. K., A. A. Rodriguez, K. S. Tsakalis. 1996. Hierarchical modeling and control of re-entrant semiconductor manufacturing facilities. In *Proceedings of the 35th Conference on Decision and Control*. Kobe, Japan.

Fowler, J. W., G. Feigin, R. Leachman. 1995. Semiconductor Manufacturing Testbed: Data Sets. Arizona State University Working Paper.

Kumar, P. R. 1994. Scheduling semiconductor manufacturing plants. *IEEE Control Systems*, 33-40.

Mason, S. J., 2000. Minimizing total weighted tardiness in complex job shops. Ph.D. Dissertation, Department of Industrial Engineering, Arizona State University.

Mason, S. J., J. W. Fowler, W. M. Carlyle. 2000. A modified shifting bottleneck heuristic for minimizing total weighted tardiness in complex job shops. Submitted to *Journal of Scheduling*.

Mason, S. J., W. M. Carlyle, J. W. Fowler. 2000. A mixed-integer program for minimizing total weighted tardiness in complex job shops. Submitted to *Operations Research*.

Pinedo, M. L., M. Singer. 1999. A shifting bottleneck heuristic for minimizing the total weighted tardiness in a job shop. *Naval Research Logistics* 46:1-17.

Uzsoy, R., C. Y. Lee, L. A. Martin-Vega. 1992. A review of production planning and scheduling models in the semiconductor industry Part I: system characteristics, performance evaluation, and production planning. *IIE Transactions: Scheduling & Logistics* 24:47-60.

Wright Williams & Kelly. 2000. Factory Explorer Version 2.7 User's Guide. Wright Williams & Kelly, California.

AUTHOR BIOGRAPHIES

SCOTT J. MASON is an Assistant Professor in the Department of Industrial Engineering at the University of Arkansas. Prior to his current position, he spent eight

years working on factory modeling, simulation, and capacity analysis projects at SEMATECH, Advanced Micro Devices, Intel, Wright Williams & Kelly, and National Semiconductor. He received his B.S.M.E. and M.S.E. from The University of Texas at Austin, and his Ph.D. from Arizona State University. His interests include modeling and analysis of semiconductor manufacturing systems, applied operations research, combinatorial optimization, and factory scheduling. He is a member of ASEE, IIE, and INFORMS. His email address is <mason@uark.edu>.

JOHN W. FOWLER is an Associate Professor in the Industrial Engineering Department at Arizona State University. Prior to his current position, he was a Senior Member of Technical Staff in the Modeling, CAD, and Statistical Methods Division of SEMATECH. He received his Ph.D. in Industrial Engineering from Texas A&M University and spent the last 1.5 years of his doctoral studies as an intern at Advanced Micro Devices. His research interests include modeling, analysis, and control of semiconductor manufacturing systems. Dr. Fowler is the co-director of the Modeling and Analysis of Semiconductor Manufacturing Laboratory at ASU. The lab has had research contracts with NSF, SRC, SEMATECH, Infineon Technologies, Intel, Motorola, PRI, ST Microelectronics, and Tefen, Ltd. He is also an Associate Editor of *IEEE Transactions on Electronics Packaging Manufacturing* and on the Editorial Board for *IIE Transactions on Scheduling and Logistics*. He is a member of ASEE, DSI, IIE, IEEE, INFORMS, POMS, and SCS. His email address is <john.fowler@asu.edu>.