5-2007

Comparison of simulation-based schedule generation methodologies for semiconductor manufacturing

Wenfeng Li
University of Arkansas, Fayetteville

COMPARISON OF SIMULATION-BASED SCHEDULE GENERATION METHODOLOGIES FOR SEMICONDUCTOR MANUFACTURING

an undergraduate Honors thesis submitted to the

Department of Industrial Engineering
University of Arkansas

by
Wenfeng Li

December 14, 2007

Mentor: Scott J. Mason Ph.D.
Reader: Edward A. Pohl Ph.D.

Abstract

Although a number of approaches have been developed to schedule tasks or jobs in many different manufacturing environments, increasing manufacturing complexity continues to motivate the need for additional scheduling heuristic research and development. This is particularly true for semiconductor manufacturing operations, arguably the most complex manufacturing environment in existence. Simulation-based scheduling has shown recent promise as a means for developing schedules for dynamic, stochastic manufacturing environments. I investigate the potential advantages and drawbacks of using simulation-based scheduling in a complex job shop as motivated by a semiconductor wafer fab.

1. Introduction

Scheduling is a decision making process used to allocate scarce resources to tasks or activities in a way such that it will meet specific objectives (Pinedo, 2002). There are many forms of scarce resources. They can be planes that are ready to take off or machines in a manufacturing facility. Tasks or activities are typically jobs that require some form of processing, like passengers who need to be flown to another airport or integrated circuits that are waiting to be processed at a particular process step. Each task may have a different priority and/or customer due date. Heuristic dispatching rules are often used to produce schedules that help companies operate more efficiently (Pinedo, 2002). Further effective scheduling techniques can save a company a significant amount of money each year.

Because of the importance of scheduling, researchers spend a lot of time creating techniques to satisfy various performance objectives. For example, the shortest processing time (SPT) rule picks the task requiring the shortest amount of time required each time the machine

becomes available. This rule is known to minimize the total time it takes to complete a set of tasks (jobs) (i.e., minimize the sum of job completion times). Besides processing time, job priority is also an important factor. Incorporating both priority and processing times, the weighted shortest processing time rule (WSPT) sorts jobs by the non-increasing ratio of job weight to processing time. However, SPT and WSPT usually do not yield acceptable solutions due to the inventory costs associated with storing jobs that finish too early. Alternatively, the critical ratio (CR) is a very popular dispatching rule in semiconductor manufacturing. The critical ratio is the ratio of the amount of time available to complete a job before its due date to the total amount of processing time that remains to be completed. The smaller a job's CR, the higher its priority becomes to a company to get it moving through the production line in order to meet its due date.

Starting in the 1960s, a new scheduling approach called simulation-based scheduling (SBS) was introduced (Wichmann 1990; Wyman 1991). This technique considers finite resources in dynamically changing factory when scheduling jobs, as opposed to previous scheduling methods, which do not consider factory dynamics (e.g., deterministic scheduling approaches). Harmonosky (1990) discusses issues of implementing simulation as a real-time decision-making tool in semiconductor manufacturing systems. Later, Harmonosky (1990) analyzes issues regarding simulation run length and types of simulation run (deterministic vs. dynamic). Soon (1997) uses simulation-based scheduling and a neutral network to solve a complex schedule. Kutanoglu and Sabuncuoglu (2001) use an experimental approach to test the effectiveness of using simulation-based scheduling to make high level decisions rather than generating complete manufacturing schedules. This SBS approach has been adopted by some manufacturing companies for its effectiveness and practicability (Wyman 1991).

In this Honor's thesis, I investigate the advantages and potential drawbacks of using SBS in a complex manufacturing environment as motivated by semiconductor manufacturing. The proposed simulation-based methodology can be described as a combination of stochastic simulation modeling and scheduling heuristics. The stochastic elements in my models include exponentially distributed product arrivals and machine break downs. I will compare SBS performance with common dispatching rules in terms of minimizing total weighted tardiness (i.e., product of job weight and *max(job completion time – due date, 0),* summed over all jobs):

- First In First Out (FIFO)—sequence jobs according to the order in which they arrive in queue
- Weighted Earliest Due Date (WEDD)—sequence jobs in non-increasing order of the ratio of job weight w_j to job due date d_j
- Critical Ratio (CR)—sequence jobs in non-decreasing order of the ratio of the amount of time left to complete a job before it's due to the total amount of process time remaining for job completion. CR is a dynamic dispatching rule, as job CR values change over time. For example, jobs currently on schedule have $CR=1$, while jobs with $CR < 1$ ($CR > 1$) are behind (ahead of) schedule.

2. Problem Description

In order to assess the advantages and potential drawbacks of SBS, I use the mini-fab model of El Adl *et al.* (1996) as the manufacturing environment under study (Figure 1). The mini-fab is used because it includes most of the essential features of a semiconductor wafer fab: re-entrant product flow, identical machines operating in parallel (tool group), and batch processing. In my research, I will not consider sequence-dependent setup times.

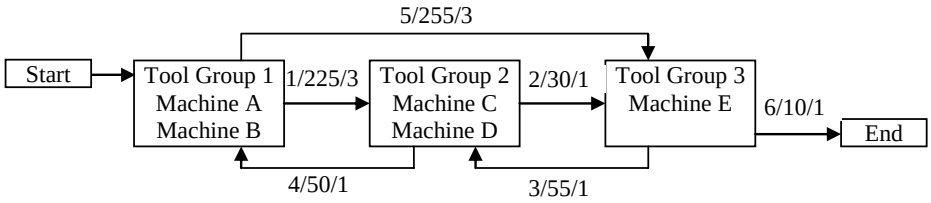

Figure 1: Five-machine six-step re-entrant mini-fab model (from Mason *et al.*, 2002)

As shown in Figure 1, the mini-fab is a six step process containing five machines (A, B, C, D, and E). The x/x/x label next to each machine in Figure 1 denotes process step number/required processing time/maximum batch size. Based on these values, the theoretical processing time of a job is 625 minutes. I assume production lots (jobs) arrive to the mini-fab according to a Poisson process, with jobs having an equal probability of being either product type 1 or product type 2. Finally each job has an associated weight or importance level and due date.

Machines A and B process jobs at Steps 1 and 5. Although both machines can process up to three lots simultaneously, a greedy batching policy allows for batches to begin processing as soon as at least one job is waiting in queue. At Step 1, jobs of both product types can be batched together. However, batch processing at Step 5 can only occur for jobs of the same product type (i.e., product-dependent batching). Machines C and D are identical and process single jobs at Steps 2 and 4. Machine E processes Step 3 and Step 6 jobs individually. A detailed description of the mini-fab's tool groups and their associated failure information is presented in Table 1. All tool failures and repairs are assumed to be distributed exponentially with the mean values presented in Table 1. These values are representative of tool failure rates in practice. As indicated in Table 1, tool groups CD and E are subject to two kinds of Failures: major and minor.

Table 1: Mini-Fab Tool Group Information

Tool Group	Machines in TG	Maximum Batch Size	Mean Time To Failure	Mean Time To Repair
AB	A, B	3 jobs	168 hours	8 hours
CD	C, D	1 jobs	Major : 300 hours Minor : 168 hours	Major : 10 hours Minor : 4 hours
E	E	1 jobs	Major : 300 hours Minor : 168 hours	Major : 10 hours Minor : 4 hours

3. SBS Model Development

Rockwell Software's Arena version 9.0 was used as the development environment for this research. Arena allows users to write custom programming logic using Visual Basic for Application (VBA) modules. I developed a VBA module for job (task) selection at each tool group so that I could embed and analyze the dispatching and scheduling rules of interest in this study.

The first two heuristic rules discussed above (FIFO and WEDD) are static in nature, as they do not depend on time. For example, jobs always get processed in the order of their arrival to each queue under FIFO. Further, under WEDD, jobs with earlier due dates and/or larger weights will always receive the highest priority, regardless of when they arrive in queue.

Once the initial Arena VBA modules were written for these static dispatching rules, the Arena model was validated by comparing my VBA code with existing dispatching rule functionality built into Arena and with existing programming code for job dispatching in static environment. Numerous cases were examined both with and without the stochastic elements present in the Arena model. Results verified proper functionality of my custom-developed Arena VBA dispatching modules.

As jobs in semiconductor wafer fabs typically have non-unit weights, I choose to implement a modified version of CR that accommodates job weights as follows:

$$CR(j,t) = \frac{d_j - t}{w_j{}^2 \left(RPT_j\right)} \quad (1)$$

In (1), $CR(j,t)$ denotes the critical ratio for job *j* at time *t,* while RPT_j denotes job *j*'s remaining processing time before completion. Clearly, this is a dynamic dispatching rule, as job *j*'s $CR(j,t)$ value is a function of time. A job is most eminent when it has the lowest $CR(j,t)$ value. Again, validation runs verified the functionality of the Arena VBA implementation of the new proposed CR rule in (1).

The CR rule in (1) is also used for scheduling within our SBS experiment. When used in a scheduling context, a job's associated $CR(j,t)$ value is only updated at fixed, pre-specified points in time (e.g., every four hours of fab time, every eight hours of fab time, etc.), rather than every time a machine is ready to process a new job (as is the case in a dispatching context). At the fixed points in time when the update occurs, both *RPT;* and time *t* are updated. Therefore, this is a dynamic dispatching rule which is updated over a specific horizon. The rolling horizon is managed using discrete event simulation.

4. Experimentation and Results

Preliminary Arena model runs identified the mini-fab's bottleneck resource to be tool group AB. In order to examine the performance of the proposed SBS approach and compare it to common dispatching methods at varying levels of fab utilization, we vary product arrival rates to the mini-fab such that the resulting bottleneck resource utilization levels were 55%, 65%, 75%, and 85% (i.e., four independent trials). We employ common random number streams for each

experimental instance in order to ensure valid comparisons between competing alternatives. All experiments were performed using Arena v9.0 installed on a 3.2 GHz PC with 2 GB of RAM.

To avoid initialization bias, preliminary analyses were performed to establish an appropriate warm-up period length for the model. Welch plots were drawn by collecting job cycle time at the 85% bottleneck utilization rate under FIFO dispatching (the case which we consider to be the baseline). It was determined that a warm-up period of 15,000 hours of mini-fab operation was sufficient to mitigate any effects of initialization bias. We ran our simulation experiments for an additional 30,000 hours to collect statistics for steady state job tardiness performance in the mini-fab under six different job selection methods:

- FIFO dispatching ("FIFO")
- WEDD dispatching ("WEDD")
- CR dispatching based on $CR(j,t)$ in (1) ("CR_Disp")
- CR scheduling, with $CR(j,t)$ values being updated every two hours ("CR_Sched_2")
- CR scheduling, with $CR(j,t)$ values being updated every four hours ("CR_Sched_4")
- CR scheduling, with $CR(j,t)$ values being updated every eight hours ("CR_Sched_8")

To better mimic reality with job due dates typically being restrictive, we set job due dates to be distributed uniformly over the interval $T_{now} + 625U[0,3]$ (where T_{now} denotes the job's creation time), while job weights were sampled from a discrete uniform distribution over the interval [1, 10].

Let $TWT(H,I)$ denote the total weighted tardiness produced by dispatching/scheduling method *H* for mini-fab model problem instance *I*. A total of five instances were created for each fab utilization level, with common random numbers being employed for each

dispatching/scheduling method H to promote valid comparisons between the competing approaches for each instance I. Let $PR(H,I) = \frac{TWT(H,I)}{\min_{H} TWT(H,I)}$ denote the performance ratio for method H in instance I. Clearly, the best method H will have $PR(H,I) = 1.000$ in each instance I. Figure 2 displays $\overline{PR(H)}$ values for each competing method H for each of the four bottleneck resource utilization levels of interest, where $\overline{PR(H)} = \frac{1}{5}\sum_{i=1}^{5} PR(H,I)$.

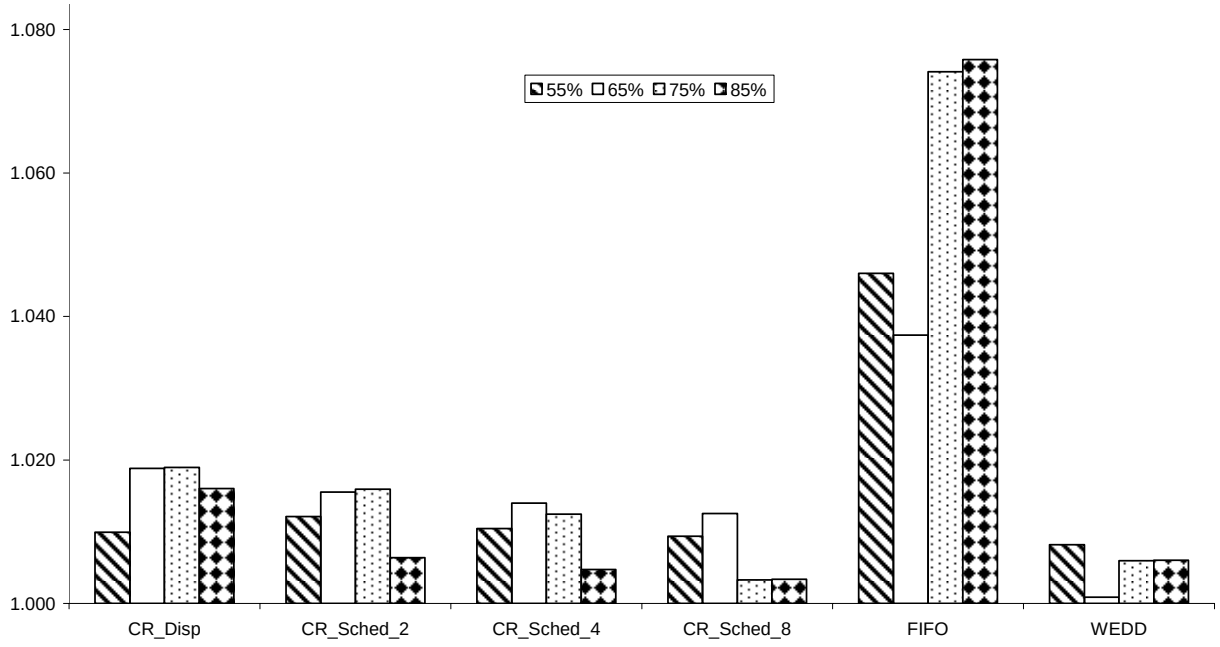

Figure 2: TWT Performance with Respect to Utilization Rates and Heuristic Rules under Exponential Release Rate

As seen in Figure 2, FIFO dispatching was the worst method at all four levels of bottleneck resource utilization. While WEDD dispatching performed the best at the two lower bottleneck resource utilization levels, CR scheduling that updates job *CR*(*j*, *t*) values every eight

hours of manufacturing time produces the best values of $\overline{PR(H)}$ for the two highest levels of bottleneck resource utilization (75% and 85%).

The reason that updating *CR(j, t)* values every eight hours provides the best results is perhaps attributed to scheduling taking advantage of its global knowledge of resources and process flows to produce better job sequences on machines as compared to dispatching. In fact, other CR_Sched rules also outperform dispatching methods at high bottleneck resource utilization levels. Though it is not always the case that WEDD outperforms CR_Sched rule, our results hold only for the experimental parameters ranges I examined. Other combinations of due dates and weights could change our results. Further research is required to determine if this scenario holds true for all instances.

Due to the dynamic nature of the mini-fab model, a sensitivity analysis was performed to gain further experimental insights. The same mini-fab model was run again with constant arrival rates that have the same output level as the previous Poisson arrival rates. The due date range was also changed to be uniformly distributed over the interval $T_{now} + 625U[1.5,3]$ thereby resulting in looser job due dates. All other system variables remained the same as in the previous run setup.

Figure 3 shows performance measures for the constant release rate cases. The best method for minimizing TWT at each utilization level changes from the variable arrival case. At 55% bottleneck utilization level, CR dispatching is the best method. However, if the bottleneck utilization level is at 65%, the best approach appears to be CR scheduling that updates a job's *CR*(*j, t*) value every eight hours. At 75% and 85% utilization level, WEDD gives the lowest $\overline{PR(H)}$. The difference in the best techniques to use under various arrival rates and due dates may be explained by the highly dynamic nature of the mini-fab model. One of the benefits

of using simulation-based scheduling is its ability to run and re-run various scenarios with very little setup time. The best methods for each scenario can be determined via simulation experimentation.

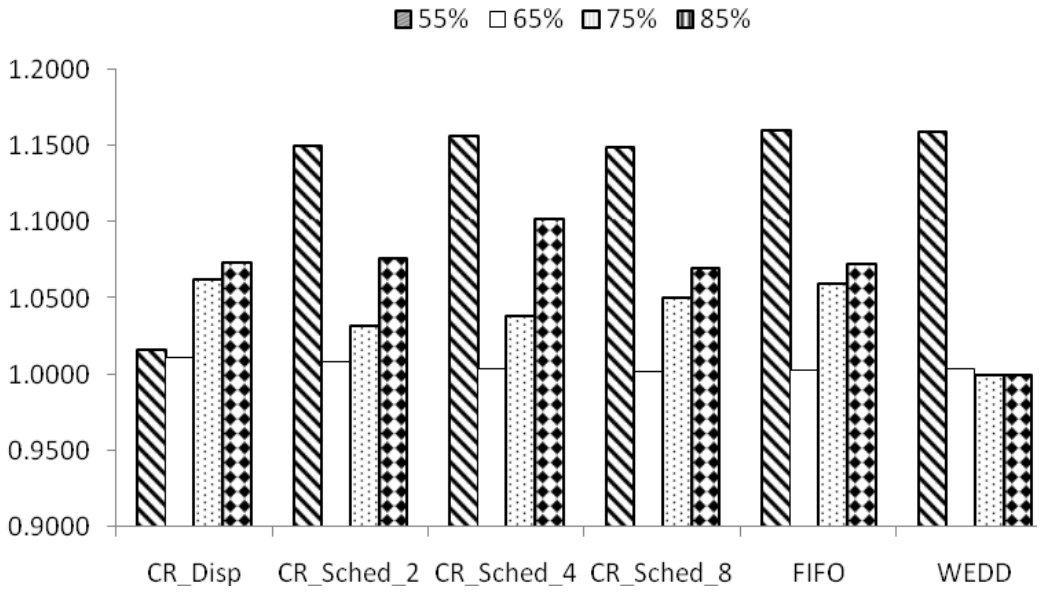

Figure 3: TWT Performance with Respect to Utilization Rates and Heuristic Rules under Constant Release Rate

5. Conclusions and Future Research

In this thesis, I investigate the advantages and potential drawbacks of using SBS in a complex manufacturing environment as motivated by semiconductor manufacturing when total weighted job tardiness is the performance measure of interest. I approach the problem by building a simulation model in Arena v9.0 and then use Arena's custom user code functionality to develop the requisite dispatching and scheduling rules in VBA modules.

Computational results show that at high production levels with exponential release rate, SBS using a proposed CR(*j, t*) method with eight hours schedule updating produces the best results with respect to total weighted tardiness. As production levels decrease, our preliminary study suggests basic WEDD dispatching best accommodates total weighted tardiness minimization. This may occur due to the method by which I set job due dates. Future research will hopefully test the proposed methodologies on actual job data from a real world wafer fab.

This effort only studied off-line simulation-based scheduling in Arena in which all job information, such as process time, due dates, and job weights, are known prior to when the process starts. For future research, an online scheduling model can be developed to test the performance of different heuristics under a rolling-horizon problem setting. Online scheduling is difficult, as jobs must be scheduled on a machine without any knowledge of future events. Finally, other future research can examine different performance measures, such as maximizing throughput or minimizing work in progress.

Acknowledgements

This research was partially supported by an Undergraduate Research Grant from the Honors College at the University of Arkansas.

References

El Adl, M. K., A. A. Rodriguez, K. S. Tsakalis, 1996, Hierarchical modeling and control of re-entrant semiconductor manufacturing facilities, *Proceedings of the 35th Conference on Decision and Control*, Kobe, Japan, 1736-1742.

Harmonosky, C. M., 1990. Implementation issues of using simulation for real-time scheduling, control, and monitoring, *Proceedings of the 22nd Conference on Winter Simulation*, New Orleans, LA, USA, 595-598.

Harmonosky, C. M., 1993. Analysis of two key issues for using simulation for real-time production control, *Proceedings of the Industrial Engineering Research Conference*, IIE, Norcross, GA, USA, 41-45.

Kutanoglu, E., Sabuncuoglu, I., 2001. Experimental investigation of iterative simulation-based scheduling in a dynamic and stochastic job shop. *Journal of Manufacturing Systems* 20 (4) 264-279.

Mason, S. J., J. W. Fowler, W. M. Carlyle. 2002. A modified shifting bottleneck heuristic for minimizing total weighted tardiness in complex job shops. *Journal of Scheduling* 5 (3) 247-262.

Pinedo, M. 2002. *Scheduling—Theory, Algorithms, and Systems*, 2nd ed. Prentice Hall, New Jersey.

Soon, T. H., 1997. Intelligent simulation-based scheduling of workcells: an approach. *Integrated Manufacturing Systems* 8 (1) 6-23.

Wichmann, K. E. 1990. Simulation-based production scheduling generation. *Production Planning & Control* 1 (3) 179-189.

Wyman, F. P. 1991. Common features of simulation based scheduling, B. L. Nelson, W. D. Kelton, G. M. Clark, eds. *Proceedings of the 1991 Winter Simulation Conference*. Phoenix, AZ, 341-347.