Proceedings of the 2001 IEEE
International Conference on Robotics & Automation
Seoul, Korea · May 21-26, 2001

An Effective Search Strategy for Wafer Fabrication Scheduling with Uncertain Process Requirements

Ming-Hung Lin and Li-Chen Fu
Dept. of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan, R.O.C
Email: lichen@ccms.ntu.edu.tw

Abstract

In this paper, we can decide the operation order with unknown potential order requirements. The scheduling architecture is proposed in this paper, First, the branch-and-bound search based on Markov chain method is proposed. The Markov chain gets the service rate records and arrival rate records from the MES(manufacturing execution system). We can get the possible beginning times of operations for each job via Markov chain. The information of the possible beginning time can help us to approximate the solution space. Thus, by the information of the possible beginning times of operations, a branch-and-bound search scheduler can be used to find a sub-optimal scheduling.

1 Introduction

Wafer fabrication is the most costly phase of semiconductor manufacturing [1, 2, 3]. Recent papers by Uzsoy et al. [2, 4], Johri [5], and Duenyas et al. [6] highlight the difficulties in planning and scheduling of wafer fabrication facilities. These papers also survey the literature on related topics. Effective shop-floor scheduling can be a major component of reduction in cycle time [7, 8, 9, 10, 11, 12, 16]. Yet in many wafer fabrications the product spends much more waiting time than actually being processed, so there is a large potential for reducing waiting time and a great benefit for doing so. People also considered other issues of wafer fabrication systems, such as batch processing system [14] and maintenance scheduling as well as staff policy [13]. It is well known in the scheduling literature that the general job shop problem is NP-hard, which lead to no efficient algorithm exists for solving the scheduling problems optimally in polynomial time for wafer fabrication.

In this paper, we assign the involved job to a machine at each wafer processing step so that the total completion time

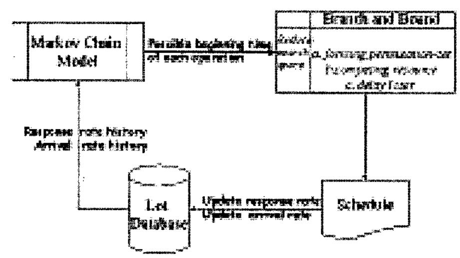


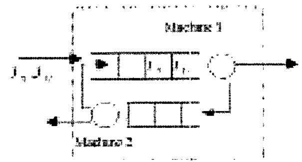
Figure 1: The Markov scheduling

of the processing is minimum subject to the constraints of some promised finishing time or due date. As we have mentioned earlier, we can get the possible beginning times of operations for each job via Markov chain. The information of the possible beginning time can help us to approximate the solution space. As shown in Figure 1, given such information, a branch-and-bound search scheduler can be used to find a sub-optimal scheduling. The Markov chain acquires the service rate records and arrival rate records from the MES (manufacturing execution system).

The organization of this paper is described as follows. In Section 2, some scheduling problems of wafer fabrication are illustrated. In Section 3, the detailed analytical Markov chain method is discussed. In Section 4, an embedded search method over the wafer fabrication system is employed. In Section 5, we demonstrate an example of using the proposed mechanism and analyze the performance. Finally, conclusions are provided in Section 6.

2 Problem Formulation

Suppose there is a set of n types of lots ($i = 1 \ldots n$). Each type has S_i wafer processing steps, and each wafer



	Step 1	Step 2	Step 3
J_{12}	Machine 1	Machine 2	Machine 1
J_{11}	Machine 1	Machine 2	Machine 1
J_{21}	Machine 1	Machine 2	
J_{22}	Machine 1	Machine 2	

Figure 2: The processing steps of 4 jobs

processing step belongs to one kind of wafer processing operation. Each processing operation is to be performed on one of the machine group, each of which has the same corresponding wafer processing capability. Let job J_{ik} be the k-th arrival lot of the i-th type, and each job J_{ik} has its arrival time a_{ik}, due date d_{ik}, and weight α_{ik}. Presume that there are totally M machines in a wafer factory. The wafer processing operation $op(l)$ of the l-th processing step of job J_{ik} requires a mean processing rate $u_{op(l),m}(i)$ on machine m. Besides, the i-th type lot is released into the wafer factory with the mean arrival rate λ_i. Let $F^l(J_{ik})$ be the time at which job J_{ik} finishes the l-th wafer processing step. Since the following discussion will all in statistical paradigm, the arrival time, processing time, and weight are modeled as independent random variables. However, we still prefer to assume that machine availability is deterministic.

Let K_i be the total number of lots of type i, which include the lots that have been released into factory and the lots that will be released. The total completion time is defined to be the weighted sum of times that all the jobs take to finish their final processing steps as shown below:

$$f = \sum_{i=1}^{n} \sum_{k=1}^{K_i} \alpha_{ik} F^{S_i}(J_{ik}) \quad (1)$$

Example 2.1

Consider the following numerical example with 4 jobs. Job J_{11} and J_{21} are the jobs corresponding to the first arrival lots of Type 1 and Type 2, respectively, whereas Job J_{12} and J_{22} are the jobs corresponding to the second arrival lots of Type 1 and Type 2, respectively. Presume that J_{11} and J_{21} have already been released into the wafer factory, but J_{12} and J_{22} have not been released into the wafer factory yet. As shown in Figure 2, from the manufacturing database Type 1 job needs three wafer operation steps and Type 2 job needs two. In this example, we suppose that the mean service time on the Machine 1 is 4 time units for each processing step of any job, and the mean service time

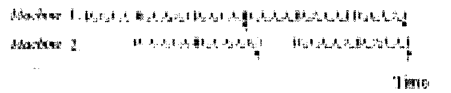


Figure 3: A possible permutation schedule in Example 1

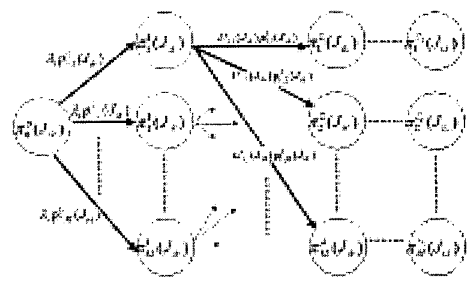


Figure 4: The Markov chain of state probability for the job J_{ik} which appears in the queue of the machine m for executing the l-th wafer processing step at time t.

on the Machine 2 is 5 time units for each processing step of any job. Moreover, we suppose that the mean arrival time for lots of Type 1 is 2 time units and the mean arrival time for lots of Type 2 is 3 time units. The scheduling problem is to determine the permutation schedule so that the total completion time f is the minimum. Figure 3 shows a possible permutation schedule where the arrows indicate the possible finishing times of the four jobs. □

3 Markov Chain Model

The job J_{ik} may be already in the factory or may wait outside the factory. Let $\pi_m^l(J_{ik})(t)$ be denoted as the state probability that the job J_{ik} appears in queue of the machine m for executing the l-th wafer processing step at time t. As shown in Figure 4, the state probability $\pi_m^l(J_{ik})(t)$ can be formulated in terms of the Markov chain. On the other hand, let $p_{s,t}^l(J_{ik})$ be the routing probability from the machine s to the machine t after the job J_{ik} finishes the l-th wafer processing step on the machine s. Let $U_{l,m}(J_{ik})$ be the mean service rate of the l-th processing step of the i-th type on the machine m.

In a practical factory, the transient solution is more meaningful than steady-state solution. Besides, there is a corresponding Markov chain as shown in Figure 4 for every job. Therefore, we need to compute the transient state probability $\pi_m^l(J_{ik})(t)$ for every job J_{ik}. Due to the special structure of this kind of Markov chain, it is possible to obtain a closed-form transient solution. We can derive a system of linear differential equations for this kind of Markov chain from differentiation and integration theory.

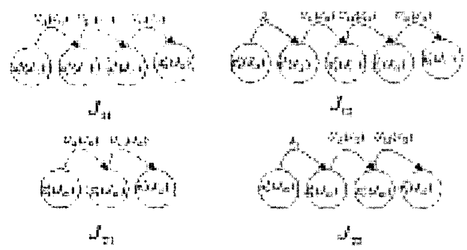


Figure 5: The Markov chains of the four jobs in Example 2.1

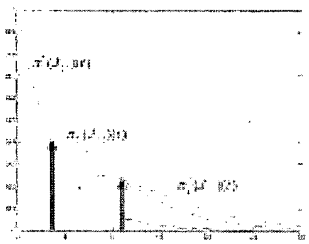


Figure 6: The transient state probability of job J_{11}

Example 3.1 *(Continued)*

Figure 5 shows the Markov chains of the jobs J_{11},J_{12},J_{21},J_{22} from Example 2.1. In this example, $\pi_0^*(J_{11})$,$\pi_0^*(J_{12})$, $\pi_0^*(J_{21})$, $\pi_0^*(J_{22})$ are the final states. We can apply elementary differentiation and integration rules and get closed-form solution for each transient state probability. The result of these transient state probabilities are shown in Figures 8 - 7 □

Precedence Constraints Considering processing-step precedence constraints, the operation *op*(l) of the l-th processing step of job J_{ik} must start before the operation *op*($l+1$) of the ($l+1$)-th processing step on the same machine m, i.e.,

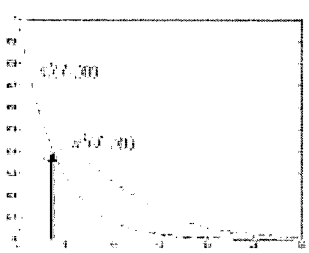


Figure 7: The transient state probability of job J_{21}

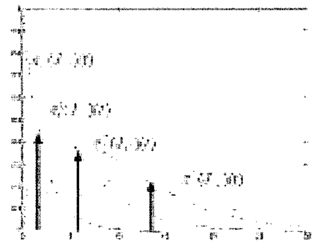


Figure 8: The transient state probability of job J_{12}

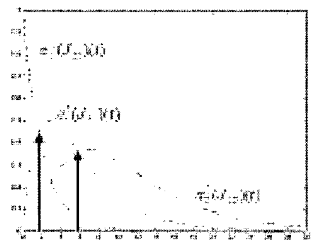


Figure 9: The transient state probability of job J_{22}

$$\pi_m^l(J_{ik})(t) > \pi_m^{l+1}(J_{ik})(t), \forall i = 1, \ldots, n, \forall k \in N$$

Thus, as shown in Figures 8- 7, we need to modify the value of $\pi_m^l(J_{ik})(t)$ where the value of $\pi_m^l(J_{ik})(t)$ is either 0 or 1 as follows, i.e.,

$$\pi_m^l(J_{ik})(t) = 1, \forall t,$$

if

$$\pi_m^{l-1}(J_{ik})(t) < \pi_m^l(J_{ik})(t)$$

or

$$\pi_m^l(J_{ik})(t) > \pi_m^{l+1}(J_{ik})(t);$$

whereas

$$\pi_m^l(J_{ik})(t) = 0, \forall t,$$

if

$$\pi_m^{l-1}(J_{ik})(t) > \pi_m^l(J_{ik})(t)$$

or

$$\pi_m^l(J_{ik})(t) < \pi_m^{l+1}(J_{ik})(t) \quad (2)$$

Resource Constraints In addition, each operation is assigned to only one machine, i.e.,

$$\sum_{m=1}^{M} \pi_m^l(J_{ik})(t) = 1, l = 1, \ldots, S_i, i = 1, \ldots, n, k = 1, \ldots, K_i \quad (3)$$

4 Branch-and-Bound Search Strategy

The scheduling problem can be viewed as finding a sub-optimal permutation schedule. We know that finding permutation schedules is NP-complete. In this paper, we use a branch-and-bound search strategy as a solution to solve such a problem in the environment of a wafer fab. It is worthwhile to note that the branch-and-bound method is based on the idea of *"intelligently"* enumerating all the feasible points in a combinatorial problem.

The qualification *"intelligent"* is important here because, as should be self-clear, it is impractical simply to examine all possible solutions. Perhaps a more sophisticated way of describing such approach is to say that we try to construct a proof to show that a solution is sub-optimal, based on successive partitioning of the solution space. The term *"branch"* in branch-and-bound method refers to the above-mentioned partitioning process, whereas the term *"bound"* refers to the lower bounds that are used to construct a proof for optimality without going through exhaustive search. We can visualize this process as a tree, where the root represents the original feasible region and each node represents a sub-problem. The natural way to branch is to choose the first job to be scheduled to each machine at the first level of the branching tree, the second job to each machine at the next level, and so on. For simplicity, let the processing operation $op(l)$ of the l-th processing step of job J_{ik} be denoted as $op_{ik,l}$. The following is an algorithm which tries to solve the scheduling problem through job assignment.

Algorithm 4.1 *The branch-and-bound search algorithm for sub-optimal scheduling:*

Begin

1. *schedule-time t := 0;*
2. *for each machine m, let PS_m be the permutation-set of machine m at schedule-time t, PS_m := 0 ;*
3. *active-set* := $\{ \underbrace{0, \cdots, 0}_{M} \}$;
4. $Q := \infty$;
5. *current-best := anything;*

For *each operation* $op_{ik,l}$ **Do**

Get mean of arrival rate and mean of processing time from manufacturing database and compute $\pi^l_m(J_{ik})(t)$;

End-for

While *active-set is not empty* **Do**

Begin

(a) *choose a branching node x, node $x \in$ active-set;*

(b) *let node x be denoted as that* $x = \underbrace{(ps_1, \cdots, ps_m, \cdots, ps_M)}_{M}$, *where* $ps_m \in PS_m$;

(c) *remove node x from active-set;*

For *each operation* $op_{ik,l}$ **Do**

If $\pi^l_m(J_{ik})(t)$=*1* **and** *operation* $op_{ik,l} \neq ps_m$ **and** $op_{ik,l} \notin PS_g$ *(subject to Equation 3)*, $\forall g \neq m$ **then** *add operation $op_{ik,l}$ into the permutation-set PS_m of machine m.*

comment : That the constraint $op_{ik,l} \notin PS_g$, $\forall g \neq m$, means that each operation is assigned to only one machine.

End-for

(d) *generate the all possible children of the node x*

(e) *child i* = $(ch_1, \cdots, ch_m, \cdots, ch_M)$, *where the winner operation* $ch_m \in PS_m$, $i = 1, \dots, n_k$, *where* $n_k = |PS_1| \times |PS_m| \times |PS_M|$;

(f) *let the loser-set LS_i of the child i be the set of loser-lists* $(ls_1, \cdots, ls_m, \cdots, ls_M)$, *where the operation* $ls_m \in PS_m - \{ch_m\}$, ls_m *is also denoted as a loser-operation competing with the operation ch_m on the machine m;*

(g) *calculate the corresponding* **lower-bound**, z_i;

For $i = 1, \dots, n_k$ **Do**

Begin

If $z_i > Q$

then *kill child i*

else $Q := z_i$, *current-best := child i, add child i to active-set;*

End

For *each loser operation* ls_m *in a loser-list*$(ls_1, \cdots, ls_m, \cdots, ls_M)$ *of loser-set* LS_i **Do**

A. *assume the winner operation ch_m is the operation $op_{ik,l}$ and* $\frac{1}{u_{op(l),m}(i)}$ *is the mean processing time of operation* $op_{ik,l}$.

B. *assume the loser operation ls_m is the operation $op_{ab,c}$ of the job J_{ab}, shift the value of the state probabilities* $\pi^c_m(J_{ab})(t)$, $\pi^{c+1}_m(J_{ab})(t)$, $\cdots$, $\pi^{S_a}_m(J_{ab})(t)$ *to satisfy the processing-step constraints of the job J_{ab};*

For processing step y **of the job** J_{ab}, y **from** c **to** S_a, **Do loop**

Begin

C. suppose t from $t1$ *to* $t2$, $\pi_m^y(J_{ab})(t) = 1$;

D. let $t0$ *be from* $t1$ *to* $t2$ *do*

Begin

E. $\pi_m^y(J_{ab})(t0 + \frac{1}{u_{op(l),m}(i)}) = \pi_m^y(J_{ab})(t0)$;

F. $\pi_m^y(J_{ab})(t0) = 0$;

End

G. suppose t *from* $t3$ *to* $t4$, $\pi_m^{y+1}(J_{ab})(t) = 1$;

H. **If** $t2 + \frac{1}{u_{op(l),m}(i)} \geq t3$ **then** *continue the loop* **else** *stop the do loop;*

End do loop

Comment : it means that the job J_{ab} *is a loser when competing with the job* J_{ik} *at its c-th processing step. Thus, all successor operations of the job* J_{ab} *may be delay caused by the job* J_{ik}*. It also indicate that all successor operations of the job* J_{ik} *may be delay by a time-period* $\frac{1}{u_{op(l),m}(i)}$ *which caused by the winner operation* $op_{ik,l}$ *of the job* J_{ik}*.*

End-for

End-for

(h) update the current schedule-time $t = t + 1$ *;*

End-while

End

5 Simulation Results

There are 24 workstations which are divided into six types, and each of the workstations comprises several identical pieces of equipment. In the simulation model presented here, each lot entering the fab is associated with a specific process flow. The model contains two different process flows. Since the process flows are deterministic, the Markov Chain can be easy constructed.

The experiment duration is 20000 hours. Eight kinds of lot-release interval are examined in this experiment, namely, 42,52,62,72,82,92,102 and 112 hours. The experiment is conducted on a computer with Intel 500 MHz CPU and 128 MB RAM. The result of this experiment is compared with the simple heuristic rule FCFS. Figure 10 shows the mean production cycle time of Type 1 lot under the cases with different lot-release intervals. We can see that the proposed branch-and-bound search based on Markov chain method produces lower mean production cycle time in comparison with the simple scheduling

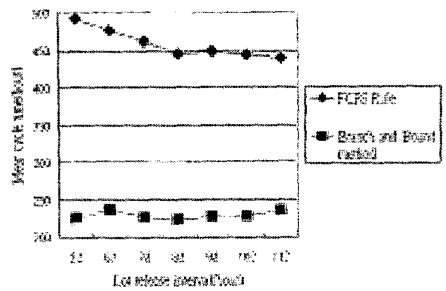


Figure 10: The mean production cycle time

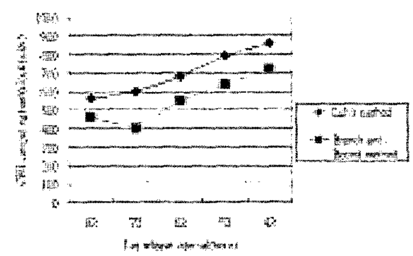


Figure 11: The computing time

based on FCFS rule. On the other hand, we provide another test case, the result of this test case is compared with Lagrangian relaxation and dynamic programming method [15]. Five kinds of lot-release interval are examined in this experiment, namely, 42,52,62,72 and 82 hours. Figure 11 shows the CPU computing time for two different scheduling strategies. We can see that the proposed branch-and-bound search based on Markov chain method needs less CPU computing time than the Lagrangian relaxation and dynamic programming method.

6 Conclusion

We can get the possible beginning times of operations for each job via Markov chain. Thus, by the information of the possible beginning times of operations, a branch-and-bound search scheduler can be used to find a sub-optimal scheduling. The results of the experiment show that the proposed branch-and-bound search based on Markov chain method produces lower mean production cycle time in comparison with the simple scheduling based on FCFS rule. In addition, the results of the experiment also show that the proposed branch-and-bound search based on Markov chain method needs less CPU computing time than the Lagrangian relaxation and dynamic programming method. Since the wafer production system is complex and the WIP level is very large, it is indeed difficult to obtain an optimal scheduling under the dynamic environment. Thus, the hereby proposed scheduling method is performed actually off-line. However, the scheduling solution by our method can be a good candidate of solution for dynamic scheduling.

References

[1] Hong Chen, J. Michael Harrison, Avi Mandelbaum, Ann Van Ackere, and Lawrence M. Wein, "Empirical Evaluation of a Queueing Network Model for Semiconductor Wafer Fabrication, " *Operation Research*, vol. 36, no. 2, pp. 202-215, 1988.

[2] Reha Uzsoy, Chung-Yee Lee, Louis A. Martin-Vega, "A Review of Production Planning and Scheduling Models in the Semiconductor Industry Part: System Characteristics, Performance Evaluation and Production Planning, " *IIE Transactions*, vol. 24, no. 4, pp. 47-60, 1992.

[3] S.M. Sze, "VLSI Technology", *McGraw-Hill* New York, 1983.

[4] Reha Uzsoy, Chung-Yee Lee, Louis A. Martin-Vega, "A Review of Production Planning and Scheduling Models in the Semiconductor Industry Part: Shop-Floor Control, " *IIE Transactions*, vol. 26, no. 5, pp. 44-55, 1994.

[5] Pravin K. Johri, "Practical Issues in Scheduling and Dispatching in Semiconductor Wafer Fabrication, " *Journal of Manufacturing Systems*, vol. 12, no. 6, pp. 474-485, 1993

[6] Lzak Duenyas, John W. Fowler, and Lee W. Schruben, "Planning and Scheduling in Japanese Semiconductor Manufacturing, " *Journal of Manufacturing Systems*, vol. 13, no. 5, pp. 323-332, 1994.

[7] Lawrence M. Wein, "Scheduling Semiconductor Wafer Fabrication, " *IEEE Transactions on Semiconductor Manufacturing* , vol. 1, no. 3, pp. 115-130, 1988.

[8] C. Roger Glassey and Mauricio G. C. Resende, "Closed-Loop Job Release Control for VLSI Circuit Manufacturing, " *IEEE Transactions on Semiconductor Manufacturing*, vol. 1, no. 1, pp. 36-46, 1988.

[9] Shu Li, Tom Tang, and Donald W. Collins, "Minimum Inventory Variability Schedule with Applications in Semiconductor Fabrication, " *IEEE Transactions on Semiconductor Manufacturing*, vol. 9, no. 1, pp. 145-149, 1996.

[10] Y. Narahari and L. M. Khan, "Modeling the Effect of Hot Lots in Semiconductor Manufacturing Systems, " *IEEE Transactions on Semiconductor Manufacturing*, vol. 10, no. 1, pp. 185-188, 1997.

[11] Yeong-Dae Kim, Jung-Ug Kim, Seung-Kil Lim, and Hong-Bae Jun, "Due-Date Based Scheduling and Contril Policies in a Multi-product Semiconductor Wafer Fabrication Facility, " *IEEE Transactions on Semiconductor Manufacturing*, vol. 11, no. 1, pp. 155-164, 1998.

[12] Yeong-Dae Kim, Dong-Ho Lee, and Jung-Ug Kim, "A Simulation Study on Lot Release Control, Mask Scheduling, and Batch Scheduling in Semiconductor Wafer Fabrication Facilities, " *Journal of Manufacturing Systems*, vol. 17, no. 2, pp. 107-117, 1998.

[13] Sarah A. Mosley, Tim Teyner, and Reha M. Uzsoy, "Maintenance Scheduling and Staffing Policies in a Wafer Fabrication Facility, " *IEEE Transactions on Semiconductor Manufacturing*, vol. 11, no. 2, pp. 316-323, 1998.

[14] C. Roger Glassey and W. Willie Weng, "Dynamic Batching Heuristic for Simultaneous Processing, " *IEEE Transactions on Semiconductor Manufacturing*, vol. 4, no. 2, pp. 77-82, 1991.

[15] Peter B. Luh, Dong Chen and Lakshman S. Thakur, " An Effective Approach for Job-Shop Scheduling with Uncertain Processing Requirements," *IEEE Transactions on Robotics and Automation* , vol. 15, no. 2, pp. 328-339, 1999.

[16] D. Y.Liao, S. C. Chang, K. W.Pei, C. M.Chang,"Daily Scheduling for R&D Semiconductor Fabrication," *IEEE Transactions on Semiconductor Manufacturing*, pp. 550-561, Nov. 1996.