Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2012, Article ID 875641, 16 pages
doi:10.1155/2012/875641

Research Article

Genetic Algorithm for Job Scheduling with Maintenance Consideration in Semiconductor Manufacturing Process

Seungchul Lee and Jun Ni

Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109, USA

Correspondence should be addressed to Seungchul Lee, seunglee@umich.edu

Received 26 March 2012; Accepted 14 September 2012

Academic Editor: Joao B. R. Do Val

Copyright © 2012 S. Lee and J. Ni. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents wafer sequencing problems considering perceived chamber conditions and maintenance activities in a single cluster tool through the simulation-based optimization method. We develop optimization methods which would lead to the best wafer release policy in the chamber tool to maximize the overall yield of the wafers in semiconductor manufacturing system. Since chamber degradation will jeopardize wafer yields, chamber maintenance is taken into account for the wafer sequence decision-making process. Furthermore, genetic algorithm is modified for solving the scheduling problems in this paper. As results, it has been shown that job scheduling has to be managed based on the chamber degradation condition and maintenance activities to maximize overall wafer yield.

1. Introduction

Semiconductors are manufactured in highly specialized facilities known as fabs. The production process in fabs usually consists of four phases: wafer fabrication, wafer probe, assembly or packaging, and final testing. Since wafer fabrication is technologically the most complex phase, its scheduling problems have been addressed by many researchers [1–4]. It is also known that the cluster tool is the basic manufacturing unit of wafer fabs [5]. Therefore, scheduling problems related to the cluster tool need to be scrutinized so that intelligent scheduling solutions that will result in the profit of companies can be obtained.

Due to the constant development of new products and processes in semiconductor manufacturing, the same equipment is very often used for different wafer processes (or recipes) [1]. However, different recipes require different operation conditions of tools on which they are executed and will thus have different influences on the tool degradation and

corresponding wafer yield [6]. Yield, defined as the percentage of working devices that emerge from the fabrication process, is undoubtedly the most important performance metric for most semiconductor fabs [6]. It is also well known that the wafer yield is highly correlated to the maintenance (or cleaning) because particulate contamination within the process equipment is a major source of yield loss [7]. Therefore, a great deal of effort in both research and industry communities has been devoted to maintenance decision making in semiconductor fabs [6, 8–12]. Yao et al. [8, 9] studied age-based preventive maintenance (PM) scheduling in semiconductor fabs. They proposed a two-level hierarchical modeling structure, which contains long-term planning as the higher level, and short-term PM scheduling at the lower level.

However, the unique characteristics of the semiconductor fabrication with multiple recipes and different degradation process are rarely seen as opportunities for helping wafer sequencing decisions while most researchers have been considering them as maintenance-related problems. For instance, it would generate higher yield to produce recipe A first when a chamber is still in good condition if recipe A is more sensitive than recipe B in terms of yield loss. Sloan and Shanthikumar [6, 10] developed a model that simultaneously determines maintenance and production schedules for a single-stage, multiproduct system. Although this model used an explicit link between equipment condition and yield loss to make optimal maintenance and production dispatching decisions, the decision-making scheme only accounted for the steady state (long-run) condition.

In this paper, the focus is placed on wafer sequencing problems considering perceived chamber conditions and maintenance activities in a single cluster tool through the simulation-based optimization method. Since this wafer release problem is too complicated to handle using analytical methods due to the highly volatile semiconductor manufacturing environment, the modified genetic algorithm (GA) whose evolutionary characteristic can provide feasible and practical solutions to both deterministic and stochastic problems is proposed to solve this problem.

The remainder of this paper is organized as follows. The problem statement and overview of the system are discussed in Section 2. Section 3 describes the framework for scheduling via simulation. Then, in Section 4, the proposed framework is validated by comparing its results with scheduling problems whose solutions are already known. In Section 5, the method is applied to wafer sequence problems in semiconductor manufacturing.

2. Problem Statement and Overview of System

We are interested in developing optimization methods which would lead to the best cassette (or batch of wafers) release policy in the chamber tool. The cassette order generated should maximize the overall yield of the wafers. Since chamber degradation will jeopardize wafer yields, chamber maintenance has to be taken into account for the cassette sequence decision-making process. For example, maintenance that is not sufficiently frequent will result in higher yield losses. On the other hand, maintenance that is too frequent will incur unnecessary maintenance costs and productivity losses. Moreover, since chamber degradation has different effects on different types of wafers [10], it would be advantageous to process the more degradation-sensitive wafers immediately after the chamber is cleaned and to process the less sensitive wafers later when the chamber condition has deteriorated.

2.1. Model for Chamber Tool Degradation

A semiconductor manufacturing system has extremely complicated processes that involve hundreds of steps and require reentrance into certain tools to facilitate manufacturing of new wafer layers. Due to the complexity of the dynamics of equipment degradation, production, and maintenance operations in semiconductor (and almost any other) manufacturing processes [13, 14], modeling of its degradation is very important for system operating point of view.

The degradation process of a chamber is modeled by a discrete-time Markov chain (DTMC) $\{X(n), n \geq 0\}$ with state space $\mathbf{S} = \{S_1, \dots, S_M\}$. A Markov chain is a stochastic process with a Markovian property, namely, that the future and past states are independent given the present state [15]. A Markov process has been widely used in optimization and control of stochastic discrete event system such as manufacturing and communication applications [6, 8–10]. Each circle in the Markov chain in Figure 1 represents a degradation state of a single chamber. The chamber goes through several states S_i, with the higher index i signifying a worse degradation level. Arrows indicate the direction of state transitions, and then the transition probabilities are depicted along with each arrow. For instance, p_{12} means the probability that S_1 will transit to S_2. In our case, the transition probabilities p_{ij} of a DTMC are zero whenever $j < i$, since the chamber condition is assumed to become only worse with time unless maintenance is performed.

Figure 1 provides an illustration of the unidirectional chamber degradation process. Whenever a maintenance action is taken (either preventive or reactive maintenance), the chamber is assumed to return to as "good as new." The established Markov models in the simulation will then enable one to track and predict levels of chamber contamination and proactively clean the chamber exactly when it is needed, rather than the current practice where chamber maintenance is based on equipment manufacturers' recommendations. Thus, the maximum usage of the tool can be exploited and a better synchronization between maintenance and production operations can be achieved.

2.2. Wafer Yield Model

The yield prediction modeling plays a crucial role in modern semiconductor fabrication [16]. Yield models can be used to determine the cost of a new chip before fabrication, identify the cost of defect types for a particular chip or a range of chips, and estimate the number of wafers required at the beginning of production [17]. In our work, we assume that the yield model for each product wafer type depends on the degree of the chamber degradation process (Markov process) when the product is processed. The yield model matrices are taken from the advanced prediction modeling presented in [17]. This yield models are generated from beta probability distributions, using a different random number seed for each matrix [10]. The values generated are sorted to ensure that yields tend to be lower as the equipment condition gets worse. The yield models are given in a form of matrix as below:

$$Y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1M} \\ y_{21} & y_{22} & & y_{2M} \\ & & \ddots & \\ y_{k1} & y_{k2} & \cdots & y_{kM} \end{bmatrix}, \tag{2.1}$$

where $y_{ij} \geq 0$ represents a yield of wafer i in state S_j and $y_{ij} \geq y_{ik}$ if $j \leq k$.

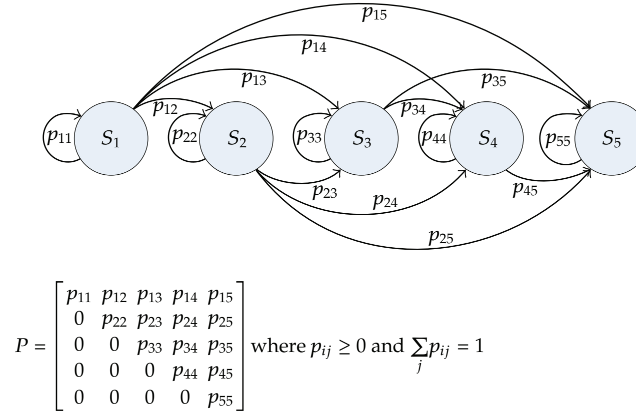


Figure 1: Illustration of state transition diagram for the Markov chain.

3. Simulation-Based Optimization For Wafer Scheduling

Most commonly used optimization procedures (e.g., linear programming, nonlinear programming, and mixed integer programming, etc.) require an explicit mathematical formulation. Such a formulation is, however, generally impossible for problems that arise in practical applications, including a chamber tool fabrication. Therefore, the approach illustrated in Figure 2 is used to address the scheduling problems in this paper [18, 19].

A feasible scheduling solution is generated by the optimization subroutine and is fed into the discrete event simulator as an input. After running the simulation with the feasible scheduling solution, the simulator yields outputs, some of which are chosen to be evaluated by an objective function. Based on the objective values calculated, the optimization subroutine produces another candidate scheduling solution given the constraint set on the inputs. These steps will be repeated until a termination condition is satisfied. The iterations are terminated if the chance of achieving significant improvement in the nest generations is excessively low.

We create the discrete event simulation with Markov processes for degradation and yield models for multiple recipes. Simulation modeling will be used to yield outputs.

3.1. Optimization Subroutine Using Genetic Algorithm

Genetic algorithm (GA) is a powerful and broadly applicable optimization search technique used to solve deterministic and stochastic problems based on the principles from the theory of evolution [20]. It is used to tackle the wafer scheduling problems in this paper as an optimization subroutine in Figure 2. Although GA is not guaranteed to generate an optimal solution, GA has shown great potential, with very promising results from experiments and practices in many different areas of the industry [21].

3.2. Modified Genetic Operations for Wafer Scheduling Problem

Since general genetic operations (e.g., crossover and mutation) in GA are not applicable to this wafer scheduling problem, special modifications to these genetic operations

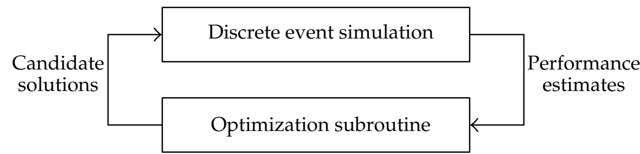


Figure 2: How scheduling optimization via simulation works.

should be made. In other words, genetic operations between two feasible sequences (parent chromosomes) do not always generate feasible sequences (offspring chromosomes). Although we can use a typical mutation operation, the crossover operations modified in this scheduling problem have to be addressed.

Suppose that 9 cassettes are in a queue, the chromosome which illustrates the job sequence of $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9$ can be represented in GA as

1	2	3	4	5	6	7	8	9

Then, the crossover operation used in this paper can be summarized as follows.

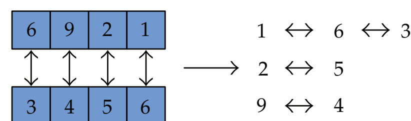
Step 1. Select two positions randomly along the chromosome

Sequence 1	1	2	3	4	5	6	7	8	9
Sequence 2	5	4	6	9	2	1	7	8	3

Step 2. Exchange two subchromosomes between sequences to create protosequences

Protosequence 1	1	2	6	9	2	1	7	8	9
Protosequence 2	5	4	3	4	5	6	7	8	3

Step 3. Determine mapping relationship between two mapping sections



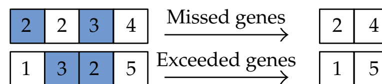
Step 4. Make updated jobs feasible with the mapping relationship

Updated sequence 1	3	5	6	9	2	1	7	8	4
Updated sequence 2	2	9	3	4	5	6	7	8	1

We also have to apply a different crossover operation for a sequence whose jobs are repeated. In this case, the following crossover operation can be conducted.

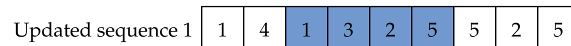
Step 1. Select two positions randomly and exchange subchromosomes

Step 2. Delete common jobs

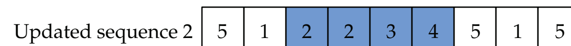


Step 3. Delete exceeded jobs from sequence 1 randomly

Step 4. Insert missed jobs into sequence 1 randomly



Step 5. Repeat Steps 2 through 4 for sequence 2 to create updated sequence 2



This special modification on the crossover operation enables one to apply the GA algorithm with a simulation to the wafer sequencing problems. The initial sequences which are randomly generated will be fed to the fabs simulation model which can provide overall wafer yield as a fitness value in GA. Based on this fitness value, GA is called again to find new sequences. These steps will be repeated until a termination condition is satisfied.

4. Validation Via Known Scheduling Problems

In this section, the methodology mentioned in the previous sections is validated through scheduling problems whose optimal sequences are already known from previous researches [22, 23].

4.1. First Example: Minimizing a Total Weighted Tardiness

Consider the following four jobs in a queue with a single machine, given the conditions in Table 1. One of the key objectives in a scheduling problem is to meet all the completion time of jobs, which, of course, depends on the schedule. The completion time of job j is denoted by

Table 1: System circumstance.

Jobs	w_j (weight)	p_j (processing time)	d_j (due date)
1	4	12	16
2	5	8	26
3	3	15	25
4	5	9	27

C_j. Then the tardiness of job j is defined as $T_j = \max(C_j - d_j, 0)$. The objective is to minimize a total weighted tardiness and it can be formulated as

$$\min_{\text{all schedules}} \sum_j w_j T_j. \tag{4.1}$$

This example of deterministic system can be found in [24] and the exact optimal schedule ($1 \to 2 \to 4 \to 3$) with an objective value of 67 is achieved via branch and bound. This problem is proved to be NP hard in the ordinary sense [24]. As shown in Table 2 and Figure 3, GA found an optimal schedule after 3 iterations.

4.2. Second Example: Minimizing a Makespan with a Parallel Machines

Consider four parallel machines and nine jobs whose processing times are given in Table 3. We want to minimize the makespan. The makespan is defined as the completion time of the last job to leave the system. Since a minimum makespan usually implies a high utilization of the machines, the makespan becomes an objective of considerable interest [25]. In addition, minimizing the makespan has the effect of balancing the load over the various machines. Therefore, the objective of this example is to minimize the makespan in the case of a four parallel machine model given the conditions in Table 3. It can be mathematically formulated as

$$\min_{\text{all schedules}} \max(C_1, C_2, C_3, C_4). \tag{4.2}$$

Some jobs have the same amount of processing time as a parallel machine configuration, prompting us to develop a different GA crossover operation (see Section 3.2). This example can be also seen in [24] and an optimal solution with an optimal value of 12 is given as $1 \to 1 \to 2 \to 4 \to 4 \to 2 \to 3 \to 3 \to 4$.

As shown in Table 4 and Figure 4, GA found an optimal schedule ($3 \to 2 \to 1 \to 4 \to 4 \to 1 \to 2 \to 3 \to 4$) after 4 iterations. Note that this optimal schedule seems to be different from the one in [24] but it turns out that this problem has multiple optimal solutions which results in the same makespan of 12.

Although we present only two numerical examples to illustrate the proposed method for solving scheduling problems, it can generally be concluded that GA works well with complex scheduling problems which are known as NP-hard problems. We want to tackle a cassette release scheduling considering machine conditions and multiple recipes in the following section.

Table 2: Simulation result sequence.

Generation	Best sequence in each generation
1	$4 \rightarrow 1 \rightarrow 2 \rightarrow 3$
2	$4 \rightarrow 1 \rightarrow 2 \rightarrow 3$
3	$1 \rightarrow 2 \rightarrow 4 \rightarrow 3$
4	$1 \rightarrow 2 \rightarrow 4 \rightarrow 3$
⋮	⋮
10	$1 \rightarrow 2 \rightarrow 4 \rightarrow 3$

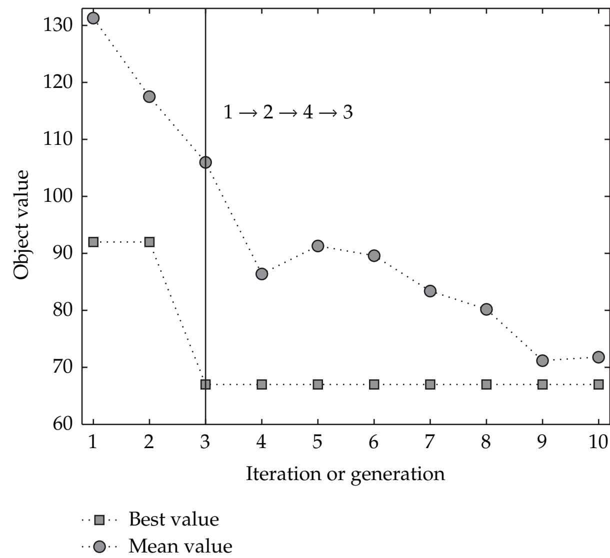


Figure 3: Simulation result: objective values.

5. Application to Semiconductor Manufacturing System

Our discrete event simulation using the ProModel software is shown in Figure 5 and can be described as follows. First, cassettes of wafers arrive in a queue at the cluster tool. Then, the cassettes of wafers are transported into the load lock by the operator. Before entering the load lock, different cassettes of wafers may be assigned different process sequences. Once the cassette has been loaded into the load lock, the cluster tool is evacuated by a pump prior to wafer processing. After the entire batch of wafers has been processed and returned to the exiting load lock, the cluster tool is raised back to the atmospheric pressure before the wafers are removed.

A set of industrial data from the semiconductor manufacturing processing has been collected from chamber tools with two different recipes (i.e., two different types of products). Nine process parameters and the succeeding metrology measurement are periodically monitored in conjunction with associated process events. Trace data sets, sampled at every second, contain current, power, gas flow, on-wafer particle counts, temperature, pressure, and so forth measured from chamber processes, while event data sets include time stamps for process and maintenance activities.

Table 3: System circumstance.

Jobs	1	1	2	2	3	3	4	4	4
p_j	7	7	6	6	5	5	4	4	4

Table 4: Simulation result sequence.

Generation	Best sequence in each generation
1	3 → 2 → 4 →1 →1 →4 →3 → 2 → 4
2	3 → 1→ 4 → 3 → 4 → 1 →2 →2 → 4
3	2 → 1 → 4 → 3 → 3 → 1→ 4 → 2 → 4
4	3 → 2 → 1 → 4 → 4 → 1 → 2 → 3 → 4
⋮	⋮
10	3 →2 →1 →4 →4 →1 →2 →3 → 4

The degradation processes for each chamber in the above cluster tool simulation have been modeled using a 5-state discrete time Markov chain as shown in Figure 1. The corresponding probability transition matrix P is obtained from a set of manufacturing process data using a hidden Markov model (HMM). Since the underlying chamber degradation condition is not directly monitored or measured, we have to estimate them by applying the HMM, addressed in [17]. An HMM enables us to estimate machine condition from a sequence of measurements (on-wafer particle counts, wafer thickness uniformity, temperature, pressure, etc.). The procedure of finding probability transition matrix from a set of industrial data via an HMM is provided in Liu [17] in detail. The resulting probability transition matrices show that recipe A requires less harsh condition than that of recipe B although the current maintenance policy does not consider this difference for the cleaning decision. Note that recipe B is more sensitive than recipe A in terms of yield loss and these characteristics are represented in the simulation model by having different values in the probability transition matrices and yield matrix as bellow:

$$
\begin{gathered}
P_{\mathrm{A}} = \begin{bmatrix} 0.9719 & 0.0195 & 0.0086 & 0 & 0 \\ 0 & 0.9256 & 0.0544 & 0.0200 & 0 \\ 0 & 0 & 0.9872 & 0.0119 & 0.0009 \\ 0 & 0 & 0 & 0.9744 & 0.0256 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \\
P_{\mathrm{B}} = \begin{bmatrix} 0.3147 & 0.6853 & 0 & 0 & 0 \\ 0 & 0.9680 & 0.0254 & 0.0066 & 0 \\ 0 & 0 & 0.7056 & 0.2806 & 0.0138 \\ 0 & 0 & 0 & 0.2279 & 0.7721 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \\
Y = \begin{bmatrix} 1 & 0.9606 & 0.8295 & 0.8122 & 0.7938 \\ 1 & 0.9224 & 0.6022 & 0.5677 & 0.4935 \end{bmatrix},
\end{gathered} \tag{5.1}
$$

where Y_{1i} = yield of recipe A in state S_i, and Y_{2i} = yield of recipe B in state S_i.

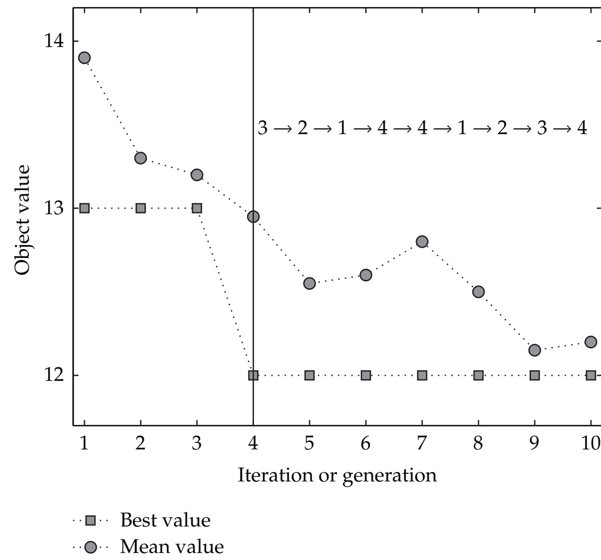


Figure 4: Simulation result: objective values.

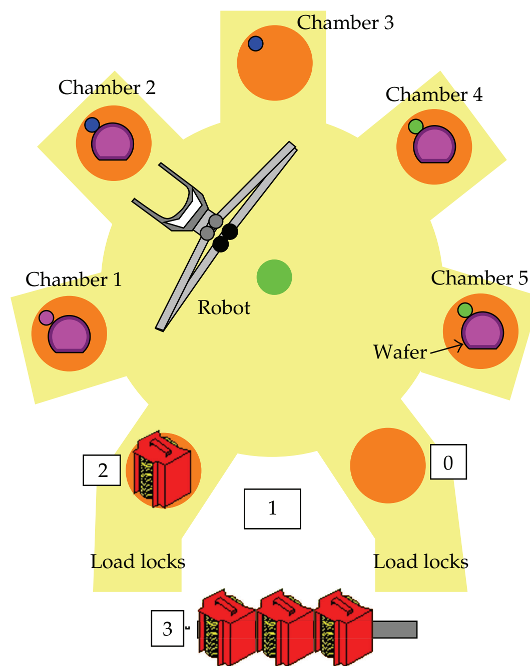


Figure 5: The layout of the cluster tool in simulation.

As mentioned in the previous section, information such as the throughput, the WIP, the yield, and the machine degradation states can all be obtained from the simulation model. We, then, use these simulation outputs to evaluate an objective function whose optimization in GA would lead to the best cassette release policy in the cluster tool.

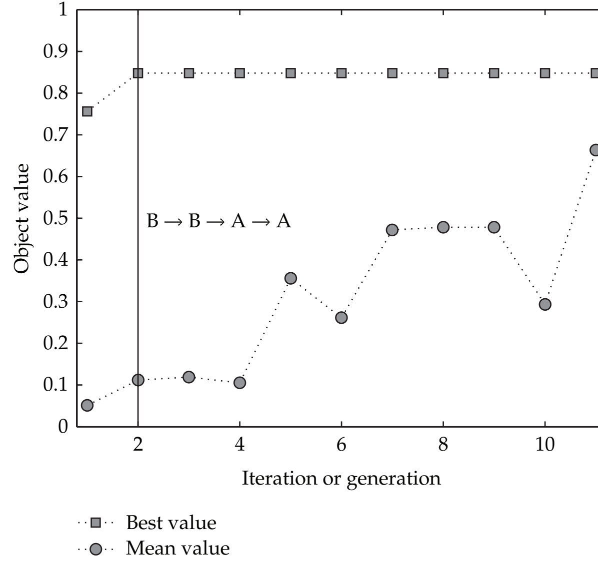


Figure 6: Objective functions over generations.

5.1. Job Release with 4 Cassettes

In this case study, two cassettes of recipe A and two cassettes of recipe B are waiting in queue. Since chamber degradation is taken into account, the scheduling of the order in which the cassettes are processed will have an effect on yield. Intuitively, recipe B will be processed first when the chambers are still in good condition because recipe B is more sensitive. This problem can be formulated as follows:

$$J = \max \frac{\sum Y_i}{n}, \tag{5.2}$$

where n is the total number of wafers processed and Y_i is the yield of wafer i.

Basically, our objective function is to maximize the average yield among the different wafers. Figure 6 and Table 5 display the simulation results obtained from GA.

GA produced the cassette sequence B $\rightarrow$ B $\rightarrow$ A $\rightarrow$ A with $J = 0.8479$. The simulation results suggest that we assign a higher priority on recipe B than on A when the chamber is in good condition, which agrees with our intuition. Since only 4 cassettes are in queue, the total processing time (makespan) is not long enough for the chamber conditions to deteriorate to the point where maintenance is required. Therefore, it is natural to process the more sensitive cassettes first and less sensitive ones later. However, this might not be the case when there are many cassettes in queue as chamber conditions might degrade to the point that maintenance is required during processing. This brings us to the next case study.

5.2. Job Release with 8 Cassettes in the Given Maintenance

In this case, we assume that there are 8 cassettes in queue so that there is a higher probability that the maintenance will be performed during processing. All parameters remain the same

Table 5: The best cassette sequences and the corresponding best objective functions.

Iterations (generations)	Best cassette sequence in population	Best objective function	Mean objective function
1	B → A → B → A	0.76	0.05
2	B → B → A → A	0.85	0.11
3	B → B → A → A	0.85	0.12
4	B → B → A → A	0.85	0.10
5	B → B → A → A	0.85	0.36
6	B → B → A → A	0.85	0.26
7	B → B → A → A	0.85	0.47
8	B → B → A → A	0.85	0.48
9	B → B → A → A	0.85	0.48
10	B → B → A → A	0.85	0.29
11	B → B → A → A	0.85	0.66

Table 6: The best cassette sequences and the corresponding best objective functions.

Iterations (generations)	Best cassette sequence in population	Best objective function	Mean objective function
1	B → B → B → B → A → A → A → A	0.52	0.21
2	B → B → B → A → A → A → B → A	0.63	0.21
3	B → B → A → A → A → B → B → A	0.73	0.04
4	B → B → A → A → A → B → B → A	0.73	0.21
5	B → B → A → A → A → B → B → A	0.73	0.55
6	B → B → A → A → A → B → B → A	0.73	0.56

except the number of cassettes which has been set to 8 in this case. The simulation results are displayed in Figure 7 and Table 6.

Here, GA produced the cassette sequence B → B → A → A → A → B → B → A with $J = 0.73$. The given cassette sequence does not seem to match our intuition. However, we find out that the maintenance occurs after the fifth job. The results suggest that we assign a higher priority on recipe B than A whenever the chamber condition is good. The recipe B is given a higher priority at the beginning of the simulation and immediately after maintenance (see Figure 8), as chamber conditions are likely to be good after maintenance. Therefore, this case study is an example to explore how cassette release and maintenance that incorporate equipment condition and yield information can influence fabs performance.

In practice, the proposed method is being implemented for (1) finding wafer sequences in more completed situation, or (2) finding the rule of thumb or insight of wafer sequences with the consideration of tool degradation, maintenance, recipes, and so forth.

5.3. Job Release with 100 Cassettes in the Given Maintenance

The previous two case studies demonstrate the importance of the joint decision for cassette sequencing and maintenance with the simplified examples. Although they provide the insight of wafer sequences with tool degradation, recipe types, and maintenance, more

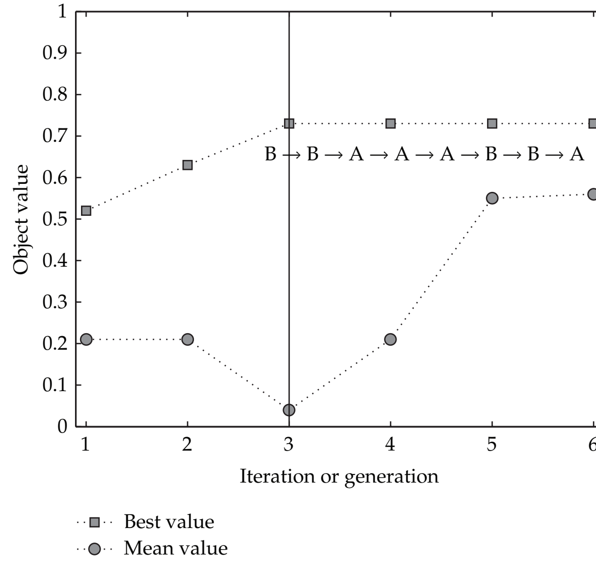


Figure 7: Objective functions over generations.

B	B	A	A	A	Maintenance	B	B	A

Figure 8: Recipe B is given a higher priority immediately after maintenance.

Table 7: Simulation results from 50 replications for each rule.

	Rule 1	Rule 2	Rule 3
Average overall yield	0.54	0.67	0.71
Standard deviation	0.04	0.03	0.04

realistic scenarios with a relatively large problem have to be investigated to show the effectiveness and feasibility of the proposed methodology. Suppose there are 50 cassettes of recipe A and 50 cassettes of recipe B in queue. 100 cassettes will be produced with the following different job dispatching rules.

(i) Rule 1: produce 50 cassettes of recipe B first, followed by 50 cassettes of recipe A.

(ii) Rule 2: produce 3 cassettes of recipe B first, followed by 3 cassettes of recipe A. Then repeat it until finishing producing all (i.e., $B \rightarrow B \rightarrow B \rightarrow A \rightarrow A \rightarrow A \rightarrow B \rightarrow B \rightarrow B \rightarrow A \rightarrow A \rightarrow A \rightarrow \cdots$).

(iii) Rule 3: produce them according to the sequence created by the proposed GA method.

Note that Rule 2 is generated according to the rule of thumb found in the second case study. As we can see in Table 7, the cassette sequencing rule generated by the proposed algorithm yields more functioning wafers than other rules. Although Rule 2 is better than Rule 1, we can conclude that job sequencing in Rule 2 is not exactly synchronized with maintenance activities during the entire production period.

Table 8: Simulation results from 50 replications for each scenario.

	Makespan		Yield	
	Scenario 1	Scenario 2	Scenario 1	Scenario 2
Average	2554.06	2636.37	0.88	0.81
Standard deviation	67.56	58.55	0.03	0.04

5.4. Wafer Dispatching

Another interesting problem is to design dispatching rules that will utilize chambers in the most efficient manner. Only recipe A will be used in this case study because we do not want the effect of changing recipes to influence our results. Suppose that the required process steps for recipe A is the following:

$$\text{chamber 1} \rightarrow \text{chamber 2} \rightarrow \text{chamber 3 or chamber 4} \rightarrow \text{chamber 5} \tag{5.3}$$

To evaluate dispatching rules on the overall yield and makespan, we will consider two scenarios of different dispatching rules which operate in the cluster tools in Figure 5.

(i) Scenario 1: a wafer will be sent to the less degraded chamber between chambers 3 and 4.

(ii) Scenario 2: a wafer will be sent to either chamber 3 or 4 randomly.

The simulation results are shown in Table 8. Shorter makespan and higher yield are achieved in Scenario 1 because the more degraded chamber which has a higher chance to produce a bad wafer is avoided. In addition, likelihood of the need for PM is lower due to the avoidance of the more degraded chamber. This can be seen in the average makespan difference of 82.31(=2636.37–2554.06) time unit of makespan. In addition, higher yield is achieved in Scenario 1 because a higher portion of the wafers are processed in the chamber which is in good condition. Therefore, this case study is an example to explore how wafer dispatching and maintenance schedules that incorporate equipment condition and yield information can affect fab performance.

6. Conclusions

In this paper, job scheduling (cassette release) with maintenance is investigated in the case of a single cluster tool in semiconductor manufacturing. It has been shown that job scheduling has to be managed based on the chamber degradation condition and maintenance activities to maximize overall wafer yield. The simulation study recommends that more sensitive wafers are to be processed whenever the chambers are in good condition. We also present the scheduling optimization methodology via simulation and this is validated using the scheduling problem whose optimal solutions are already known. This approach can be used for more complex scheduling problems. For genetic algorithm, the crossover operation is modified for solving the wafer scheduling problems in this paper.

Acknowledgments

This work is partially supported by the Industry/University Cooperative Research for Intelligent Maintenance Systems (NSF Grant no. 0639468) and Semiconductor Research Corporation (SRC) Task ID 1222.003.

References

[1] R. Uzsoy, C. Y. Lee, and L. A. Martinvega, “A review of production planning and scheduling models in the semiconductor industry part I: system characteristics, performance evaluation and production planning,” *IIE Transactions*, vol. 24, no. 4, pp. 47–60, 1992.

[2] R. Uzsoy, C. Y. Lee, and L. A. Martin-Vega, “Review of production planning and scheduling models in the semiconductor industry part II: shop-floor control,” *IIE Transactions*, vol. 26, no. 5, pp. 44–55, 1994.

[3] Y. H. Lee, B. K. Lee, and B. Jeong, “Multi-objective production scheduling of probe process in semiconductor manufacturing,” *Production Planning and Control*, vol. 11, no. 7, pp. 660–669, 2000.

[4] P. Jula and R. C. Leachman, “Coordinated multistage scheduling of parallel batch-processing machines under multiresource constraints,” *Operations Research*, vol. 58, no. 4, part 1, pp. 933–947, 2010.

[5] W. M. Zuberek, “Cluster tools with chamber revisiting—modeling and analysis using timed Petri nets,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 17, no. 3, pp. 333–344, 2004.

[6] T. W. Sloan and J. G. Shanthikumar, “Combined production and maintenance scheduling for a multiple-product, single-machine production system,” *Production and Operations Management*, vol. 9, no. 4, pp. 379–399, 2000.

[7] P. G. Borden and L. A. Larson, “Benefits of real-time, in situ particle monitoring in production medium current implantation,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 2, no. 4, pp. 141–145, 1989.

[8] X. Yao, M. Fu, S. I. Marcus, and E. Fernandez-Gaucherand, “Optimization of preventive maintenance scheduling for semiconductor manufacturing systems: models and implementation,” in *Proceedings of the IEEE International Conference on Control Applications (CCA ’01)*, pp. 407–411, September 2001.

[9] X. Yao, E. Fernández-Gaucherand, M. C. Fu, and S. I. Marcus, “Optimal preventive maintenance scheduling in semiconductor manufacturing,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 17, no. 3, pp. 345–356, 2004.

[10] T. W. Sloan and J. G. Shanthikumar, “Using in-line equipment condition and yield information for maintenance scheduling and dispatching in semiconductor wafer fabs,” *IIE Transactions*, vol. 34, no. 2, pp. 191–209, 2002.

[11] A. S. Charles, I. R. Floru, C. Azzaro-Pantel, L. Pibouleau, and S. Domenech, “Optimization of preventive maintenance strategies in a multipurpose batch plant: application to semiconductor manufacturing,” *Computers and Chemical Engineering*, vol. 27, no. 4, pp. 449–467, 2003.

[12] A. Crespo Marquez, J. N. D. Gupta, and J. P. Ignizio, “Improving preventive maintenance scheduling in semiconductor fabrication facilities,” *Production Planning and Control*, vol. 17, no. 7, pp. 742–754, 2006.

[13] J. Hunter, D. Delp, D. Collins, and J. Si, “Understanding a semiconductor process using a full-scale model,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 15, no. 2, pp. 285–289, 2002.

[14] T. Freed, K. H. Doerr, and T. Chang, “In-house development of scheduling decision support systems: case study for scheduling semiconductor device test operations,” *International Journal of Production Research*, vol. 45, no. 21, pp. 5075–5093, 2007.

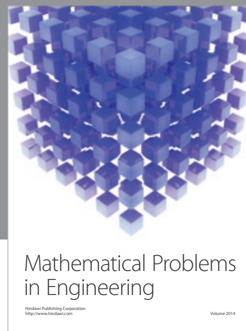
[15] S. M. Ross, *Stochastic Processes, Probability and Statistics*, John Wiley & Sons, New York, NY, USA, 2nd edition, 1996.

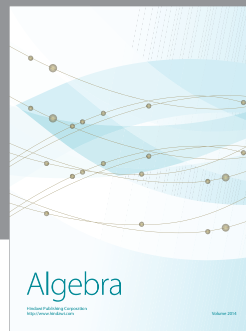
[16] J. Moyne and B. Schulze, “Yield management enhanced advanced process control system (YMeAPC) part I: description and case study of feedback for optimized multiprocess control,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 23, no. 2, pp. 221–235, 2010.

[17] Y. Liu, *Predictive modeling for intelligent maintenance in complex semiconductor manufacturing processes [Ph.D. thesis]*, University of Michigan, Ann Arbor, Mich, USA, 2008.

[18] M. C. Fu, “Optimization via simulation: a review,” *Annals of Operations Research*, vol. 53, no. 1, pp. 199–247, 1994.

[19] M. C. Fu, "Optimization for simulation: theory versus practice," *INFORMS Journal on Computing*, vol. 14, no. 3, pp. 192–215, 2002.
[20] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Design*, John Wiley & Sons, New York, NY, USA, 1997.
[21] Z. M. Yang, D. Djurdjanovic, and J. Ni, "Maintenance scheduling for a manufacturing system of machines with adjustable throughput," *IIE Transactions*, vol. 39, no. 12, pp. 1111–1125, 2007.
[22] L. Min and W. Cheng, "Genetic algorithms for the optimal common due date assignment and the optimal scheduling policy in parallel machine earliness/tardiness scheduling problems," *Robotics and Computer-Integrated Manufacturing*, vol. 22, no. 4, pp. 279–287, 2006.
[23] I. Essafi, Y. Mati, and S. Dauzère-Pérès, "A genetic local search algorithm for minimizing total weighted tardiness in the job-shop scheduling problem," *Computers and Operations Research*, vol. 35, no. 8, pp. 2599–2616, 2008.
[24] M. L. Pinedo, *Scheduling Theory, Algorithms, and System*, Prentice Hall, Englewood Cliffs, NJ, USA, 2008.
[25] L. Min and W. Cheng, "A genetic algorithm for minimizing the makespan in the case of scheduling identical parallel machines," *Artificial Intelligence in Engineering*, vol. 13, no. 4, pp. 399–403, 1999.





Hindawi

Submit your manuscripts at
http://www.hindawi.com