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

Modeling, Scheduling, and Prediction in Wafer Fabrication Systems Using Queueing Petri Net and Genetic Algorithm

Hung-Wen, Li-Chen Fu, and Shih-Shinh Huang

Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan, R.O.C.

Abstract--Wafer fabrication is one of the most competitive manufacturing business in the world. In order to survive from such strongly competitive environment, finding an effective schedule which can result in higher machine utilization and throughput rate, shorter cycle time, and lower WIP (Work-In-Product) inventory becomes a major task. Besides that, in order to help customers to make ordering decisions as well as to let the manager control the processing conditions of the fab, we need to predict some performance measures efficiently. In this paper, we propose a modeling tool named Queueing-Petri Net (Q-PN) which combines the characteristics of Queueing Theory and Petri Net. It can be used to model various detail of the manufacturing system as well as to get its performance evaluation very efficiently. Then, a general Q-PN model is presented to simulate the semiconductor manufacturing system. Based on this model, we propose a Genetic Algorithm (GA) based scheduler and an analysis-based predictor. In the GA scheduler, the chromosome represents a combination of scheduling policies, including lot release policies, machine selection rules, dispatch rules and batch rules. So, when the GA finishes its optimization process, an optimal scheduling policy is produced. As for the predictor, because it inherits the analytical property of Queueing Theory from the Q-PN model, we can use it to predict those performance measures efficiently such as the exact due date of some particular lot.

1. Introduction

Wafer fabrication is the most costly phase of semiconductor manufacturing [3]. A significant amount of risk is involved in the wafer fabrication because it requires huge investment costs and complex manufacturing process. To survive from such competitive and risky environment, the company must not only reduce production cycle time and increase throughput rate but also meet customers' due dates.

Recent papers by Uzsoy et al. [3,5] and Johri [4] highlight the difficulties in planning and scheduling of wafer fabrication facilities. These papers also survey the literature on releated topics. Effective shop-floor scheduling can be a major component of reduction in cycle time. The benefits of effective scheduling include higher machine utilization, shorter cycle time, higher throughput rate, and greater customer satisfaction. This is particularly true of semiconductor manu-facturing, with its rapidly changing markets and complex manufacturing processes [1,6,7,8]. 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. 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, and therefore it is the reason why we apply the genetic algorithm (GA) to approach the problem. GA is a search procedure that uses random choices as a guide tool through a coding in the parameter space [9-13]. While randomized, however, GAs are by no means a simple random-walk approach. They efficiently exploit historical information to speculate on new search nodes with expected improved performance. That is, GA samples large search space randomly and efficiently to find a good solution in polynomial time, which however does not require enormous memory space as other traditional search algorithms such as A*. Many researchers have used GA to deal with job shop scheduling problem in traditional industries. Lee *et al.* [12] focused on solving the scheduling problem in a flow line with variable lot sizes. Lee *et al.* [11] combined the machine learning and genetic algorithm in the job shop scheduling. Ulusoy *et al.* [13] have addressed on simulateneous scheduling of machines and automated guided vehicles (AGVs) using genetic alogrithm. In addition, Cheng *et al.* [10] have surveyed relational topics on solving the job shop problem using GA. They also discussed chromosome representation in details. Unlike the previous research, we use GA methodology to solve the more complex scheduling problem in wafer fabrication. However, since wafer fabrication is a complex discrete event system, schedulers cannot be easily realized on this kind of system, and thus how to model a complex wafer fab manufacturing system is a imperative task. In the modeling field, Petri Net (PN) has played an important role; it has been developed into a powerful tool for discrete event systems, particularly in manufacturing systems. PNs have gained more and more attentions in semiconductor manufacturing due to their graphical and mathematical advantages over traditional tools to deal with discrete event dynamics and characteristics of complex systems [14-17]. Zhou *et al.* [15] reviewed applications of PNs in semiconductor manufacturing automation. It can also serve as a tutorial paper. The colored timed Petri net (CTPN) is used to model the furnace in the IC wafer fabrication [17]

and in the entire wafer fabrication manu-facturing system [14]. Jeng *et al.* [16] applied Petri net methodologies to detailed modeling, qualitative analysis, and performance evaluation of the etching area in an IC wafer fabrication system located in the Science Based Industrial Park in Hsinchu, Taiwan.

In this paper, we propose a systematic color-timed Petri-Net (CTPN) model embedded with a genetic algorithm (GA) scheduler. The CTPN can be used to model the complex process flows in wafer fab efficiently and the detailed manufacturing characteristics. Also, new transitions are introduced in this paper, which significantly reduce the complexity of Petri-Net model. The GA scheduler can be used to dynamically search for an appropriate dispatching rule for each workstation or processing unit family through the CTPN model.

The organization of this paper is described as follows. In Section 2, the definitions of the proposed color-timed Petri net (CTPN) are revealed here. And, the systematic method of CTPN model is discussed. In Section 3, a GA embedded search method over the CTPN model is employed. In Section 4, we demonstrate two example of using the proposed mechanism and analyze the performance. Finally, conclusions are provided in Section 5.

2. Wafer Processing Model

The ordinary PN do not include any concept of time and color. With this class of nets, it is possible only to describe the logical structure and behavior of the modeled system, but not its evolution over time and color. Responding to the need to model the manufacturing system in wafer fab, we add time and color attributes to the ordinary PN. In the proposed colored-timed Petri-Net (CTPN) model, we introduced three kinds of places, namely immediate (ordinary) places, timed places, and communication places, and five kinds of transitions, i.e., immediate (ordinary) transitions, colored transitions, mapping transitions, comparable transitions, and macro transitions. Also, we have introduced colored tokens in the CTPN model. For clarity, the tokens' color and transitions are described as follows.

Tokens' color

In the CTPN model, the color of the tokens is defined as a 5-digits number. The first two digits are defined as the product type, i.e., the route ID, and the last three digits are defined as the operation step that the product is performed now.

Immediate transitions

Immediate transitions are the same as the ordinary transitions. They can be used to model behaviors or events of resources in manufacturing systems.

Colored transitions

There are a set of colored transitions T_c and a set of color C in CTPN model. For all $t \in T_c$, which contains a color set $C(t)$. We define that $t \in T_c$ can be enabled with respect to the color c if

$$m(p,c) \geq I(p,t,c), \quad \forall p \in P, c \in C(t);$$

where

P is the set of places in CTPN model.

$m(p,c)$ is the number of tokens in p with respect to the color c.

$I(p,t,c)$ is the multiplicity of input arcs from p to t with respect to the color c.

After the colored transition t is fired, the new marking m' becomes

$$m'(p,c) = m(p,c) + O(p,t,c) - I(p,t,c).$$

In the CTPN model, the colored transition is drawn as ▬.

Mapping transitions

The function of the mapping transitions is to transfer the token's color c_1 to a predefined color c_2, i.e., after firing the mapping transitions, the color of the tokens that enable this type of transition is transferred to the predefined color of other kind. The other enabling and firing rules of the mapping transitions are the same as the ordinary transitions. In the CTPN model, the mapping transitions are drawn as ▬.

Comparable transitions

A comparable transition has two input arcs and two output arcs, which was drawn as ⊠. One of the comparable transition's output arcs is a regular arc; the other one is an inhibitor arc. Only one of the comparable transition's output arcs can be enabled. The mechanism of the comparable transitions is that, the comparable transition compare the two token's color in the two input places, if the colors are the same, the regular output arc is active; otherwise, the inhibitor arc is active.

Macro transitions

Each macro transition contains a module of the model, and the module contains a set of subnets. In the CTPN model, the macro transitions are drawn as ▭.

In this paper, we used the proposed CTPN to model the whole wafer manufacturing systems which include the deposition, photolithography, etching, and diffusion areas. The wafer processing model we proposed is a general model, which does not focus on some special cases. In other words, when the equipment information is given, the wafer processing model is automatically generated by the model generator. Different process flows of different products can be performed based on this model by changing the tokens' color, as long as these process flows were performed in the fab using the proposed CTPN model. The implementation of the model generator was discussed in Section 4, and the details of the proposed Wafer Processing Model including Routing Module and Elementary Module (see Figure 2) are described in the following sections.

The purpose of the Routing Module is to model the logical process flow of the manufacturing systems. The basic concept of this module is described as fol-

lows. First, we divide all the machines in the fab into *n* workstations (machine groups or processing unit family), each of which contains one or more identical machines (or processing units). At the beginning, a token (lot) with the color xy000 enters the model, where xy is the route ID, and 000 is current operation step. Lots then checkin and the lots' color are changed to xy001, preparing to do the first operation. Each operation has its associated workstation to be performed, thus lots travel to and take operations in the proper workstation in the fab according to the predefined process flow. After the lot finishes the current operation, the lot' s color number will be increased by one, and ready to do the next operation. Step by step, after the lot finishes all its operations, it enters the *end* place and finishes the work. The Routing Module which is shown in Figure 3 implements the idea above-mentioned. For clarity, we explain the notations used in Figure 3 as follows:

The purpose of Operating Module is to model the detailed manufacturing system in a wafer fab, such as processing, setup, rework, scheduled machine maintenance as well as unscheduled machine breakdown, and time-critical operation. We divide the Elementary Module into two subnets, which will be explained in details as follows.

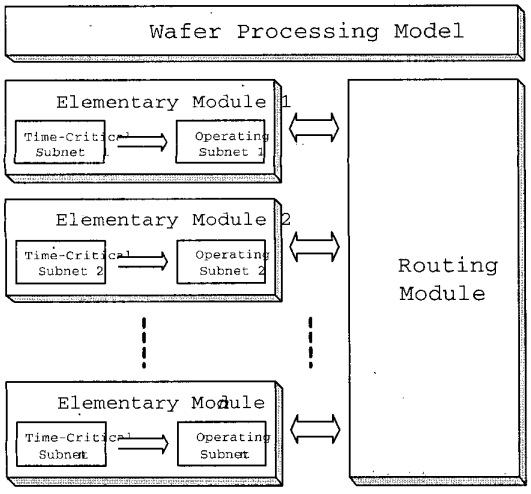

Figure 2 Wafer processing model

Time-Critical Subnet

Time-Critical Subnet is shown in Figure 4. This subnet is used to avoid the rework in some time-critical operations. The idea of the Time-Critical Subnet is that if a lot is waiting for a time-critical operation, after waiting for some specific time period, the lot can get the higher priority and can be performed first. The notation used in Figure 4 is described below.

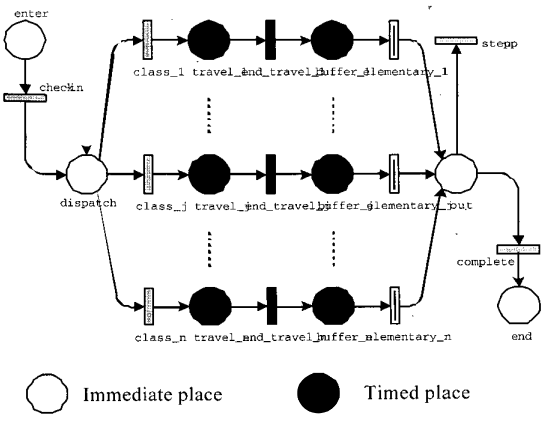

Figure 3 Routing module.

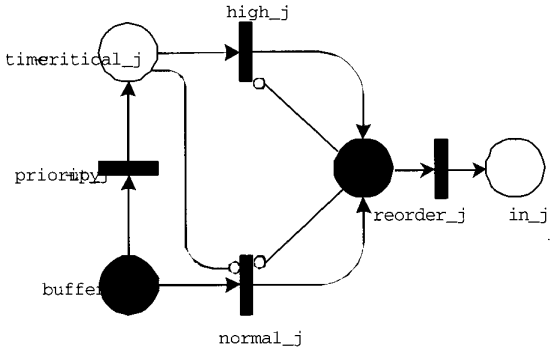

Figure 4 Time-Critical Subnet.

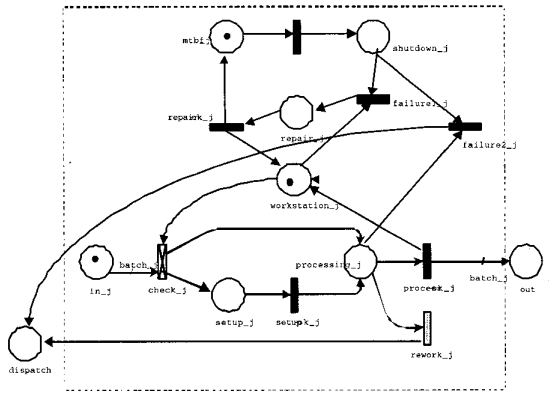

Figure 5 Operating Subnet

Operating Subnet

Operating Subnet is shown in Figure 5. The purpose of this subnet is to model processing, machine setup, machine failure, and to check whether the lot needs to be reworked. Similarly, we explained the notation used in Figure 5 as follows.

3. Wafer Fab Scheduler

In this paper, we allow our computation model to support the search algorithm over a color-timed place Petri-Net (CTPN) model, i.e., search can be performed in both axes of multiple resources and different time segments. Here, we propose a new scheme to represent a schedule for the problem of production scheduling in wafer fab using GA embedded search over a CTPN model. The algorithm starts with an initial set of random configurations called a population, which is a collection of chromosomes. The chromosome here denotes a total scheduling solution for wafer fabrication. The size of the population is always fixed (N_p). Following this, a mating pool is established in which pairs of individuals from the population are chosen. The probability of choosing a particular individual for mating is proportional to its fitness value. Chromosomes with higher fitness values have a greater chance of being selected for mating. Applying crossover (R_c) to generate new offsprings. Mutation and inversion are also applied with a low probability. Next the offsprings generated are evaluated on the basis of fitness, and selecting some of the parents and some of the offsprings forms a new generation. The above procedure is executed N_g times, where N_g is the number of generations. After a fixed number of generations (N_g), the fittest chromosome, i.e., the one with the highest fitness value is returned as the desired solution.

Chromosome Representation

In this paper, we use priority rule-based representation of chromosomes in the GA. This representation belongs to indirect approaches as described above, which brings us the advantages such as the simplicity of the chromosome structure, simple GA operators, and shorter computation time. First, we define a gene place as follows:

P_g : A *gene* place set is a subset of the place set, i.e., $P_g \subset P$. A gene place $p \in P_g$ is used to control the scheduling in GA over CTPN model.

The gene place in our CTPN model is denoted as the input buffer place of each machine group, that is, each machine group (identical machines) has a gene associated with it. A gene $g = (d, s, b)$ is a three tuple where

- d: one type of dispatching rules.
- s: one type of setup rules.
- b: one type of batching rules.

The rules we selected for gene codes are listed in Table 1.

Table 1 Gene codes

Dispatching Rules		Set-up Rules		Batching Rules	
Name	Code	Name	Code	Name	Code
FCFS	1	FCFS	1	W1T	1
MINS	2	SSU	2	W2T	2
SRPT	3			W3T	3
EDD	4			W4T	4

For each rule, it is described as follows:

- **FCFS**: First Come First Serve.
- **MINS**: Minimum Inventory at the Next Station first. In this rule, a lot has a higher priority if its next operation workstation has a lower inventory.
- **SRPT**: Shortest Remaining Processing Time first
- **EDD**: Earliest Due Date first
- **SSU**: Same Set-Up first
- **W*x*T**: Waiting for the arrival lot to complete the batch within x unit of lot inter-arrival time. When the batch is completed within this specific time period, the batch is started immediately. Otherwise, the partial batch is started right after one unit of lot inter-arrival time.

After genes are defined, the chromosome can be created. In this paper, the length of a chromosome is fixed, and is equivalent to the number of machine groups. The structure of the chromosome is depicted in Figure 6.

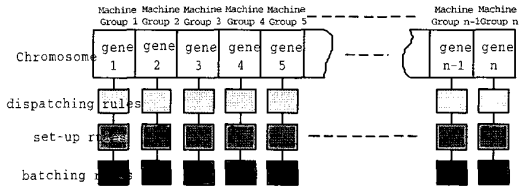

Figure 6 Chromosome representation

Fitness Function

In this paper, we use three objective functions in our implementation. The fitness function is defined as follows:

$$f(c) = w_1 \cdot f_1(c) + w_2 \cdot f_2(c) + w_3 \cdot f_3(c)$$

where f_1 is the score for mean cycle time,

f_2 is the score for throughput rate,

f_3 is the score for number of lots which meet due date.

Scheduler Builder

A schedule builder is dedicated to transform a chromosome to a feasible schedule, such that we can evaluate the aforementioned indirect chromosome representation. Based on the CTPN model, the evolution of the system can be addressed by the change of marking in the net. Consequently, all possible kinds of behavior of the system can be completely tracked by the reachability graph of the net. In other words, we can track the WIP status from the CTPN model while the schedule was performed. Thus, given a CTPN model and a chromosome, the schedule builder can generate a feasible schedule in terms of the firing sequence of transitions in the CTPN model according to the chromosome. The firing sequence of transitions

provides the order of the initiation of operations. The architecture of the scheduler was shown in Figure 7, in which, we first select a lot release policy to control the timing for release of a lot (token) to the CTPN model. Second, apply the GA over the CTPN to find a good chromosome. Finally, use the schedule builder to transform the chromosome to a feasible schedule.

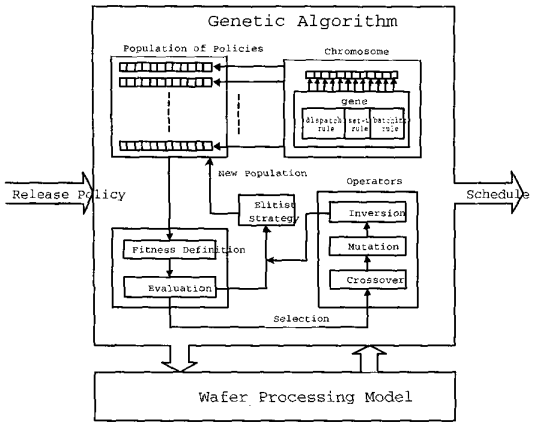

Figure 7 The architecture of the scheduler

4. Experiment Results

The simulation model we used in this paper is based on Wein's model [1]. It describes a fictitious wafer fab, but most of the parameters of the model are derived from data gathered at the Hewlett-Packard Technology Research Center Silicon fab (the TRC fab), which is a large R&D facility in Palo Alto, California. Unlike the model studied by Wein [1], we enlarge the capacity of the fab to increase the complexity of the simulation. Moreover, we add the rework probability to the inspection machines, and define the reworked step in each route. Batch processing machines are also included in our simulation model, so that the simulation model can be closed in real fab. We assume both of the workstations of TMNOX and PLM6 are batch-processing machines, and their batch size are 4 and 2, respectively. The entire fab contains 24 workstations (total 74 machines) in our simulation model, and three different process flows are defined. Route 1 has 172 steps, and its total processing time is 494.6 hours. Route 2 has 139 steps, and its total processing time is 412.7 hours. Route 3 has 110 steps, and its total porcessing time is 346.7 hours.

Case 1: Three Orders for Total 100 Lots

The problem description is listed in Table 2

Table 2 Three orders

Order ID	Route	Quantity (lots)	Due Date
Order 01	Route 1	30	1999/12/18 08:00
Order 02	Route 2	20	2000/01/03 08:00
Order 03	Route 3	50	2000/01/13 08:00

Current System Date: 1999/10/30

We evaluated the performance of the five scheduling policies, which are FCFS, SRPT, MINS, EDD, and GA. In addition, the four simple dispathcing rules plus SSU are also evaluated. For each scheduling policy, we run 10 times of simulation and calculate mean value and standard deviation. The the compared results are listed in Table 3, where the three criteria are mean queueing time (MQT), throughput rate (TPR), and the rate of meeting due date (MDD).

Table 3 The compared result for case 1

Item	MQT (hours)		TPR (lot/day)		MDD (%)	
	Mean	Std-Dev	Mean	Std-Dev	Mean	Std-Dev
FCFS	467.5	7.5	2.197	0.030	69.9	3.0
FCFS+SSU	626.6	67.2	2.019	0.065	45.1	9.5
SRPT	472.1	16.5	1.955	0.031	74.8	2.8
SRPT+SSU	545.3	31.1	1.963	0.024	58.4	7.9
MINS	458.1	12.1	2.180	0.026	71.7	3.8
MINS+SSU	626.9	61.6	2.031	0.047	43.7	9.2
EDD	498.0	18.2	2.020	0.034	65.8	6.8
EDD+SSU	627.8	62.2	1.952	0.085	38.5	10.6
GA	374.1	4.8	2.296	0.014	91.5	1.0

Case 2: Four Orders for Total 80 Lots

Unlike the case 1, we have four orders in the case 2. Also, the route sequence we used as the input pattern in the case 2 is different from the case 1. Here we listed the four orders, and one compared result in Table 4, and Table 5, respectively.

Table 4 Four orders to be released into the fab

Order ID	Route	Quantity (lots)	Due Date
Order 01	Route 2	20	1999/12/12 08:00
Order 02	Route 3	20	1999/12/17 08:00
Order 03	Route 1	10	1999/12/24 08:00
Order 04	Route 3	30	2000/01/04 08:00

Current System Date: 1999/10/30

Table 5 The compared result for case 2

Item	MQT (hours)		TPR (lot/day)		MDD (%)	
	Mean	Std-Dev	Mean	Std-Dev	Mean	Std-Dev
FCFS	441.5	5.8	2.224	0.022	62.7	2.7
SRPT	487.8	57.3	2.036	0.056	74.6	3.2
MINS	439.2	8.4	2.223	0.020	62.5	2.6
EDD	472.0	10.7	2.107	0.035	50.0	3.3
GA	371.5	5.2	2.325	0.023	83.0	2.3

From Table 3 and Table 5, we found that the proposed GA scheduler performs much better than other conventional dispatching rules. It has a lower queueing time for lots spent in the fab, a higher throughput rate for a fab, and a higher rate for meeting the customers' due date. In addition, the experimental results show that the proposed GA scheduler has a lower variability on the total queueing time, throughput rate, and the rate of meeting due date, which increases the

accuracy of the simulation based prediction. As a result, the proposed GA scheduler has a significant impact on wafer fab scheduling, by providing obvious improvements over the other conventional dispatching rules, even though the fab has a mixed production.

5. Conclusion

In this paper, we consider the wafer fab scheduling problem. We first proposed a systematic colored-timed Petri-Net (CTPN) model for a wafer fab. The entire CTPN model is composed of two modules, one is Routing Module, and the other is Elementary Module. The objective of the Routing Module is to model the logical process flow of the wafer fab manufacturing system. And, the Elementary Module is used to model the detailed manufacturing characteristics in wafer fab. In this paper, we also introduced many new transitions, which are useful to model some special issue and to significantly reduce the complexity of Petri-Net model in our study. In order to make better scheduling policies on wafer fab, we proposed a genetic algorithm scheduler, which dynamically searches for the appropriate dispatching rules for each machine group or processing unit family. Through the experiments, we found that the GA scheduler provides more superior performance than the conventional dispatching rules do. By using GA scheduler, we have a higher throughput rate for fabs, a shorter queueing time for lots spent in the fab, and a higher promising rate for meeting the customers' due date.

Reference

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] Sadiq M. Sait and Habib Youssef, *VLSI Physical Design Automation: Theory and Pratice*, McGraw-Hill, New York, 1995.

[10] Runwei Cheng, Mitsuo Gen, and Yasuhiro Tsujimura, "A Tutorial Survey of Job-Shop Scheduling Problems Using Genetic Algorithm, Part I: Representation," *Computers and Industrial Engineering*, vol. 30, no. 4, pp. 983-997, 1996.

[11] C. -Y. Lee, S. Piramuthu, and Y. -K. Tsai, "Job Shop Scheduling with a genetic algorithm and machine learning", *International Journal of Production Research*, vol. 35, no. 4, pp. 1171-1191, 1997.

[12] In Lee, Riyaz Sikora, and Michael J. Shaw, "A Genetic Algorithm-Based Approach to Flexible Flow-Line Scheduling with Variable Lot Sizes", *IEEE Transactions on Systems, Man, Cybernetics—Part B: Cybernetics*, vol. 27, no. 1, pp. 36-54, 1997.

[13] Gunduz Ulusoy, Funda Sivrikaya-Serifoglu, and Umit Bilge, "A Genetic Algorithm Approach to the Simultaneous Scheduling of Machines and Automated Guided Vehicles," *Computer Operations Research*, vol. 24, no. 4, pp. 335-351, 1997.

[14] M. H. Lin and L. C. Fu, "Modeling, Analysis, Simulation, and Control of Semiconductor Manufacturing Systems: A Generalized Stochastic Colored-Timed Petri-Net Approach," *IEEE International Conference on Systems Man, and Cybernetics*, 1999.

[15] MengChu Zhou and Mu Der Jeng, "Modeling, Analysis, Simulation, Scheduling, and Control of Semiconductor Manufacturing Systems: A Petri Net Approach," *IEEE Transactions on Semiconductor Manufacturing*, vol. 11, no. 3, pp. 333-357, 1998.

[16] Mu Der Jeng, Xiaolan Xie, and Shih Wei Chou, "Modeling, Qualitative Analysis, and Performance Evaluation of the Etching Area in an IC Wafer Fabrication System Using Petri Nets," *IEEE Transactions on Semiconductor Manufacturing*, vol. 11, no. 3, pp. 358-373, 1998.

[17] Sheng-Ya Lin and Han-Pang Huang, "Modeling and Emulation of a Furnace in IC Fab Based on Colored-Timed Petri Net," *IEEE Transactions on Semiconductor Manufacturing*, vol. 11, no. 3, pp. 410-420, 1998.