Dedicated Machine Constraint: Using a Multiagent Scheduling System

Abstract

We present a Multiagent Scheduling (MS) system to tackle the dedicated machine constraint. The dedicated machine constraint is one of the new issues of the photolithography machinery due to natural bias. Natural bias will impact the alignment of patterns between different photolithography layers. The dedicated machine constraint is the most important challenge to improve productivity and fulfill the request for customers in semiconductor manufacturing today. In this paper, the proposed MS system is based on a Resource Schedule and Execution Matrix (RSEM) and keeps the load balancing among photolithography machines during each scheduling step according to the current load among the photolithography machines in the production line. Along with an example, we describe the prototype system including the agents and the coordination strategies in the paper. We also demonstrate the simulation result that validated the proposed MS system.

Introduction

The dedicated machine constraint is one of the new issues of the photolithography machinery due to natural bias in semiconductor manufacturing. Natural bias will impact the alignment of patterns between different layers. The smaller the dimension of the IC products, the more difficult they will be to align between different layers, especially when we move on to a smaller dimension IC for high technology products. The dedicated machine constraint is one of the challenges in the semiconductor manufacturing systems.

Semiconductor manufacturing systems are different from the traditional manufacturing operations, such as a flow-shops manufacturing system in assembly lines or a job-shops manufacturing system. In a semiconductor factory, one wafer lot passes through hundreds of operations, and the processing procedure takes a few months to complete. The operations of semiconductor manufacturing incrementally develop an IC product layer by layer. A "Re-Entrant" production line has been proposed to model the process flow of a semiconductor manufacturing system (Kumar 1993) (Kumar 1994). A research for semiconductor reentrant manufacturing has been proposed a hierarchical approach (Vargas-Villamil, Rivera, and Kempf 2003). Figure 1 shows the concept of the operation flows of a semiconductor manufacturing system.

The load balancing issue of photolithography machines in the semiconductor factory is derived mainly from the dedicated machine constraint. This happens because once the wafer lots have been scheduled to one of the machines at the first photolithography stage, they must be assigned to the same machine in the subsequent photolithography stages until they have passed the last photolithography stage. If we randomly schedule the wafer lots to arbitrary photolithography machines at the first photolithography stage, then the load of all photolithography machines might become unbalanced. The unbalanced load of photolithography machines is the most important challenge to improve productivity and fulfill the request for customers in semiconductor manufacturing today. It is also the main contributor to the complexity and uncertainty of the semiconductor factory as well.

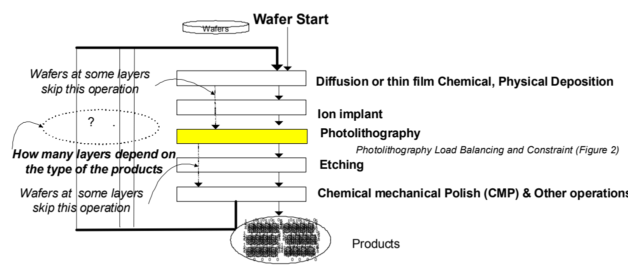


Figure 1: The operations of semiconductor manufacturing

A study concerning the load balancing issue developed a load balance allocation function by applying a dynamic programming method to the machine constraint in the photolithography machines (Miwa, Nishihara, and Yamamoto 2005). Arisha and Young proposed a Neural Network approach to improve the performance of the photolithography machines (Arisha and Young 2004). Mönch, et al. worked on the wafer lots assignment of the photolithography machines to improve the load balancing problem (Mönch, Prause, and Schmalfuss 2001).

In this paper, motivated by the problems above, we proposed the Multiagent Scheduling (MS) system based on a Resource Schedule and Execution Matrix (RSEM) to tackle the dedicated machine constraint and load balancing issue (Shr, Liu, and Chen 2006a) (Shr, Liu, and Chen 2006b). The RSEM has been proposed in the research as a representation method for the tasks and the manipulation of execution steps.

Most of the recent research or projects related to the manufacturing industry have mentioned the issue that traditional centralized control systems and scheduling models could not handle the complexity and uncertainty of, nor the rapidly changing market for, modern manufacturing production. Semiconductor manufacturing is a typical example with the features of internal unpredictable machine breakdown and external fast-changing market demand. A considerable amount of research has been carried on recently on these topics, such as ADACOR (Leitão, Colombo, and Restivo 2005), a distributed shipboard water-cooling system (Maturana, Staron, and Hall 2005), the intelligent manufacturing systems (Maˇrík and McFarlane 2005), and ExPlanTech (Pˇechouˇcek, Vokˇrínek, and Beˇcváˇr 2005).

The proposed MS system keeps the load balancing among photolithography machines during each scheduling step. The MS system is to schedule the wafer lots at the first, the unconstrained, photolithography stage to the suitable photolithography machine according to, not only the scheduling priority of the wafer lots, but also the current load among photolithography machines in the semiconductor manufacturing system. Agents collect the current information such as the counts of wafer lots in the queue buffers of machines, the load status of each photolithography machine, and so on from the RSEM. Then they cooperatively execute the scheduling task to achieve load balancing among photolithography machines with the dedicated machine constraint.

The rest of the paper is organized as follows. First, we present the proposed MS system. Second, we describe the prototype system including the agents and the coordination strategies. After that, we will demonstrate the simulation result that validated our proposed approach. We conclude the research with discussion and our intended future work.

Multiagent Scheduling System

In the Multiagent Scheduling (MS) system, agents can communicate with each other asynchronously, negotiating and cooperating to achieve the goal without a central decision component. Researchers who have been attempting to apply agent-based technology to complex manufacturing problems have developed a concept, the holonic manufacturing system (Van Leeuwen and Norrie 1997). The holonic manufacturing system consists of holons which are autonomous units and can communicate with other holons in the manufacturing system. Some of the holons will report to other holons, some of the holons will cooperatively achieve a goal, and some of them will negotiate with each other for an agreeable solution to conflicting goals. Moreover, the most important feature of the holonic manufacturing system is that holons must be replaceable with each other in case of breakdown. It is also similar to agent technology (Koestler 1967).

Before scheduling a wafer lot to a dedicated machine at the first photolithography stages, the MS system considers the load, utilization and some other factors of the photolithography machines at the same time. Load and utilization can be calculated by the following equations.

$$\textit{Load (m)} = \frac{\sum\{n(w) \mid w \in W, Ph(w) = m\} \times R(w)}{WPH(m) \times T}$$

$$\textit{Utilization (m)} = \frac{\sum\{n(w) \mid w \in W, Ph(w) = m, S_k(w) < T\}}{WPH(m) \times T}$$

m: photolithography machine
W: wafer lots in process (WIP)
n(w): wafer quantity of w
Ph(m): $w \in W$, limited to the photo machine m
R(w): remaining photolithography stages for wafer w
T: time duration; T usually will be a 24-hour day.
$S_k(w)$: slack time for wafer w
WPH(m): wafer per hour, the productivity of machine m.

Load is defined as the rate of the wafer lots limited to machine *m* multiplied by the remaining layers of photolithography stage (demand) to the capacity of the machine *m* (provide). Load is a relative parameter, representing load statue among photolithography machines. The larger load factor means that the more required service from wafer lots has been dedicated to this machine. Utilization is defined as the rate of the wafer lots limited to machine *m* and capacity of machine *m*. When the utilization of the machine *m* is greater than 1, then the machine *m* is overloaded and some of the wafer lots waiting for machine *m* will not be processed within time period *T*. Wafer lots will take more time to exit the system than the original plan. This delay situation usually could be made up by a scheduling policy, Fluctuation Smoothing Policy for Mean Cycle Time (FSMCT) (Kumar and Kumar 2001), at next following non-photolithography stages. However, if the machine is overloaded all the time, the wafer lots limited to the machine will have serious delay, that is the total queue time for these wafer lots will far more than the standard queue time for these wafer lots. Therefore, they will be never on time for delivery, even if these wafer lots could always get the highest priority for the following stages.

The wafer lots that have been scheduled to the machine will compete with other wafer lots in the buffer queue limited to this machine. Wafer lots in the following photolithography stages will be queued in the buffer of each photolithography machine and follow FSMCT scheduling policy. Wafer lots in non-photolithography stages will be queued in the buffer for all groups of machines following FSMCT scheduling policy as well.

There are three type of agents in the MS system: one *Plan agent* represents the factory's production planner to set up production goals for the factory according to the orders by customers; one *Schedule agent* represents a photolithography scheduler, to schedule the all wafer lots at the first photolithography stage to the suitable photolithography machine; and many photolithography *Machine agents* represent photolithography machines. Agents are to follow a coordination protocol similar to bit framework protocol to cooperatively balance the load of photolithography machine.

A scenario could be as follow: *Schedule agent* deals with the wafer lots and generates the requirements to meet the delivery time set by the *Plan agent*. On behalf of photolithography machines, the *Machine agents* issue a proposal to bid the task according the load and utilization factors of the machines. There are some additional factors to evaluate the task proposal issued by the *Machine agent*. Some of them represent the machine status and the others the service levels which the machines can provide. The system combines these factors to evaluate the capability of the machine to the one wafer lot and determines a “best fit” machine for it. The Schedule agent will announce the requirements for all the wafer lots at the first photolithography stages that have not been scheduled or need to be re-scheduled due to serious machine breakdown, or an abnormal event to the system. Then the *Machine agents* will propose the "ability" of the machine they represented for these requirements. After the *Schedule agent* reviews all these machine abilities proposed by the Machine agents, these wafer lots will be scheduled to the best machine. All these three above activities process repeatedly until no wafer lots are needed to be scheduled.

Schedule agent will initiate the scheduling process under two conditions. One is that new wafer lots have arrived at the first photolithography stage. Once the scheduling process has started, the wafer lots that have not been assigned to any machines will be scheduled to one dedicated photolithography machine. The other condition is that some special event has happened, such as a serious machine breakdown, or a process issue that prohibits some photolithography machines from any more processing of a customer's products. These wafer lots will be rearranged by *Plan agent* and rescheduled by *Schedule agent* to suitable photolithography machines again.

We have normalized the value of all factors as between 0 and 1, and the weights of factors could be turn on (1) or turn off (0) which depend on factory status.

X_1: Machine capacity, the productivity of a machine within a given time period and the value depends on the performance, Wafer per Hour (WPH), the value of this factor will be 0: below the average, 0.5: approximate the average, 1: better than average.

X_2: Machine loading, Load is the total service demand of the wafer limited to this machine to the capacity of the machine could provide.

X_3: Machine utilization, it represents the workload percentage of machine in one day and is a short-term factor.

X_4: How fast the wafer lot could be served? The priority of the wafer lot is in the queue of the machine or the estimated waiting time for service is by this machine. It follows FSMCT, the LS policy.

X_5: Machine yield of this machine for some customer's products. The value of this factor will be 0: below the average, 0.5: approximate the average, 1: better than average.

X_6: Mask on Machine, according to family based scheduling rule, if the wafer will use the same mask, the machine could save the setup time cause by process change in the machine.

X_7: Machine Status, the status value will be running (Run), idle (Lost), Periodical Maintenance (PM) or breakdown (Down). The value for status as follow: 1 for Run, 1 for Lost, and 0 for Down. The value of PM will depend on the estimated remaining time of the operation and the value is 1-(remaining time/24).

$$X_{best\text{-}fit} = \sum a_i X_i, i = 1\sim7,$$
$$\text{Turn on} \rightarrow a_i = 1, \text{ Turn off} \rightarrow a_i = 0$$
$$\text{wafer lot is scheduled to the machine } m \text{ with } MAX(X_{best\text{-}fit})$$

To prevent the situation that no one wants to bit, we have regulated *Machine agents* such that every photolithography machine has to enter the bit if the following conditions are all true:

- *the load of the machine is below the average,*
- *the machine has no constraint to the wafer lot, which needs to be scheduled to a photolithography machine, and,*
- *the machine is idle.*

The system combines these factors to evaluate the ability of the machines to the wafer lots. *Machine agents* respond to the *Schedule agent's* requirements in the system, and then provide the factors, $X_1 \sim X_7$, to the system according the each wafer lot which is going to be scheduled that can reduce much effort and computation time. While the wafer lots have been assigned to the best suitable machine, the parameters of the wafer lots will be updated and put into the queue buffer of the machine. *Plan agent* could change the priority of the wafer lot in LS scheduling policy. For example, *Plan agent* extends the due date of the wafer lot which will never be on time for delivery. *Plan agent* can also slowdown or speedup some wafer lots by modifying the due data according to the customer's special request.

Prototype System

In this section, our system design of the MS system is presented. We adapted the multiagent system developed by Lou, et al. (Lou, Shr, and Hu 1998) and follow the format of message to build up the MS system. Figure 2 shows the data structure and system design of the proposed MS system. *Schedule agent* starts every scheduling cycle from it issues a contract for the wafer lot in the wafer queue buffer, and then it inquires and get the bits from *Machine agents;* the final step is to select a best machine for the wafer lot. After scheduling the wafer lot, the contract will be put into the contract list. These three steps cooperating with *Product agent* and *Machine agents* are repeated by *Schedule agent* until there is no wafer lot in the wafer queue.

Agents

Schedule agent has a priority wafer queue for the wafer lots at the first photolithography stage and a contract list for all the contacts. The wafer lots in the wafer queue are set by *Product agent*. Each contact is to record the assigned machine ID, wafer lot ID and the machine's factors. The current average load of the machines is updated by *Schedule agent* before starting every bit process. We could learn how to adjust the weight of factors from the history of the contacts and the average load at that time point. *Plan agent* consists of scheduling list and product list. The wafer lots at the first photolithography stage are arranged into the scheduling list by their priority. The product list has the product's information such as customer ID, due date, and completed rate. The wafer lots of the product are in the product list. Each wafer lot has the wafer's information including ID, constrained photolithography machine, priority, due date, and the quantity.

Machine agent has the information of the photolithography machine, including performance index, WPH, and current status, running, idle, and so on. The number of *Machine agents* will be the same number as the photolithography machines the semiconductor factory has. *Machine agent* will update the load, utilization, and other factors when any event happens to the machine, such as a wafer lot having completed a stage process, or a new wafer lot has been scheduled to this machine. *Machine agents* also have a stage list to know which stages the machine can serve and a queue buffer for the wafer lots waiting for this machine. The constrained wafer lot list is to record all wafer

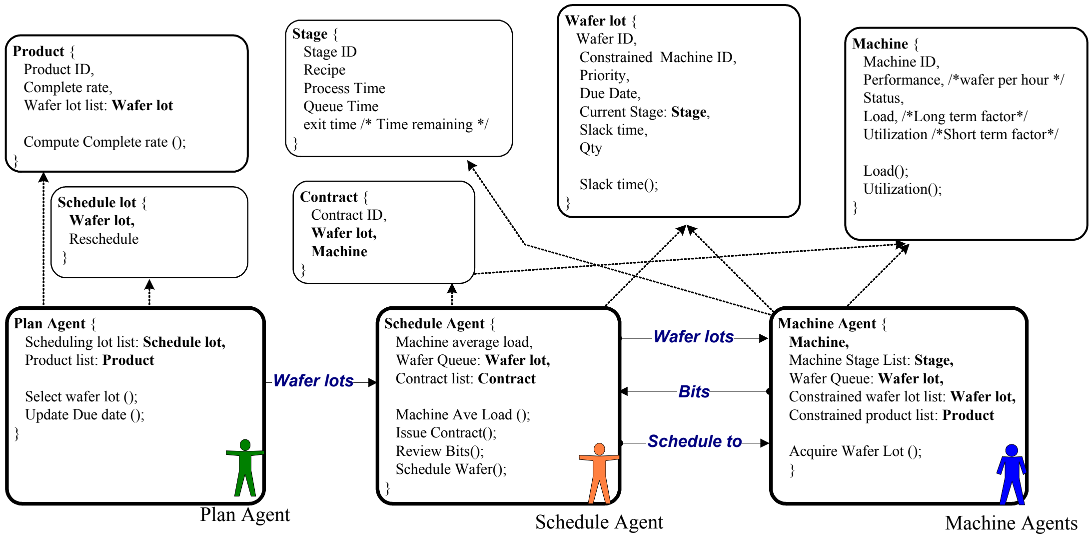


Figure 2: Agents

lots of the production line dedicated to this machine, and the constrained products list is to record the products that the machine cannot process. Therefore, according to the information, *Machine agents* can determine when to join the bit for acquiring more wafer lots for the machine.

Resource Schedule and Execution Matrix (RSEM)

In the semiconductor factory, the tasks are the procedures of processing wafer lots, starting from the raw material until the completion of the IC products. We use the task matrix of the Resource Schedule and Execution matrix (RSEM) as a tool to representation the temporal relationship between the wafer lots and machines. The RSEM consists of the following:

- *a task matrix Λ,*
- *a set of factors generating from the task matrix Λ, and*
- *a set of predefined rules.*

A task matrix Λ. The task matrix includes a finite set $\sigma = \{S_1,\ldots, S_n\}$ *of* n steps and a finite set $\tau = \{T_1,\ldots ,T_m\}$ *of* m tasks. Each S_i has a resource capacity vector $c_i = [c_{ik}]_{1xh}$; each T_j requires a resource vector $r_j = [r_{jk}(t)]_{1xh}$, i.e., a sequential pattern to represent the resources it needed during the process sequence from a raw material to a product. Therefore, the task matrix Λ looks as follows:

	S_1	S_2	.	.	.	.	.	S_j	.	.	S_m
T_1	r_1	r_2	r_3	..	..	..	r_k.	..	..	..	..
T_2		r_3	r_4	..	..	r_k	..	..	..	..	..
.							..	..			
.				r_1	r_3			r_k	..		
T_i					r_3	r_4	..	..	r_k	..	
.							.	.			
T_n					..	..	r_k	..	..	..	

The symbol r_k in the matrix Λ is to represent the task T_i needs the resource (or machine) r_k at the time S_j. If T_i starts to be processed at S_j, we will fill its pattern into the matrix from $[T_i, S_j]$,... *to* $[T_i, S_{j+p-1}]$, i.e., p *is* the total step numbers of T_i. All the tasks, σ, follow the illustration above to form a task matrix Λ in the task generation module. To represent the dedicated machine constraint in the matrix for this research, the symbol r_k^x, a replacement of r_k, is to represent that T_i has been dedicated to number x of type k machine at S_j. The symbol w_k is to represent the wait situation when the machine r_k cannot serve T_i at S_j.

A set of factors generating from the task matrix Λ. The factors are summarized the value of each dimension as the scheduling rules of the MS system. For example, we can acquire how many steps T_i needed to be processed by counting task pattern of T_i dimension in Λ. We can also realize how many wait steps T_i has had by counting w_k from start step to current step of T_i dimension in Λ. Furthermore, if we count the r_k^x in S_j dimension, we can know how many tasks will need the machine m_x of r_k at S_j. The definitions and formulae of these factors are as follows:

Required resource (machine):

- *How many tasks will need the resource r_k at S_j?*

$$RR(r_k^x, S_j) = \sum_{T_i \in W} \Lambda[T_i, S_j] = r_k^x, 1 \le x \le p$$

,p is the number of resource k.

- *How many tasks will wait for the resource r_k at S_j?*

$$QueueBuf(r_k, S_j) = \sum \Lambda[T_i, S_j] = w_k$$

Count steps of tasks:

- *How many wait steps T_i has had before current step?*

$$WaitStep(T_i) = \sum_{j=start}^{current\ step} \Lambda[T_i, S_j] = w_k, 1 \le k \le o$$

,o is the number of resource types.

- *How many steps t_i will have?*

$$Count(T_i) = \sum_{j=start}^{end\ step} \Lambda[T_i, S_j] \ne ""$$

A set of predefined rules. After the system generates the task matrix Λ and obtains all the factors for the scheduling rules, and build up the rules. The proposed MS system starts to schedule the wafer lots to the suitable machine following the factors and regulations defined. To represent the situation of waiting for r_k; i.e., when T_i can not take the resource r_k at S_j, then we will not only insert w_k in the pattern of T_i, but also need to shift the following pattern to the next step in the task matrix $\boldsymbol{\Lambda}$. For example, at S_{10}, T_i has been assigned to m_1, therefore, T_{i+1} will have a w_2 be inserted into at S_{10}, and then all the following required resource of T_{i+1} will shift one step. The following matrix shows the situation. All the other types of machines will have the same process, but do not need to be concerned with the dedicated machine constraint. Therefore, we assigned one of the wafer lots which has the largest WaitStep(T_i), then the second largest one, and so on, for each machine (resource r_k). Similarly, the MS system will insert a w_k for the wafer lots do not be assigned to machines r_k. The factor, WaitStep(T_i), is to represent the delay status of T_i.

$$
\begin{array}{lccccccccccc}
 & & S_9 & S_{10} & S_{11} & S_{12} & S_{13} & S_{14} & .. & .. & S_j & .. \; S_m \\
 & & & & & & .. & .. & & & & \\
T_i & .. & \boxed{\boldsymbol{r_2}^{\boldsymbol{1}}} & \boldsymbol{r_4} & r_5 & r_6 & r_7 & .. & .. & & & \\
T_{i+1}{}' & .. & \boxed{\boldsymbol{w_2}} & \boldsymbol{r_2}^{\boldsymbol{1}} & \boldsymbol{r_4} & r_6 & r_5 & .. & .. & .. & .. & \\
.. & & \uparrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & & & & &
\end{array}
$$

Discussion. We assume that all the resource types for the wafer lot will have the same process time in RSEM, i.e. all the steps have the same time duration. The assumption simplifies the real semiconductor manufacturing system. However, it is not difficult to approach the real world on a smaller scale time step. Another issue is that the machines in the factory have capacity limitation due to the capital invention, which is the resource constraint. How to make the most profit for the invention mostly depends on optimal resource allocation techniques. However, most scheduling polices or methods can neither provide the exact allocation in accepted time, nor a robust and systematic resource allocation strategy. We use the RSEM to represent the complex tasks for the MS system to allocate resources by the simple matrix calculation. This reduces much of the computation time for the complex problem.

The RSEM could provide two kinds of functions. One is that we can follow the predefined rules from expert knowledge to obtain the resource allocation result at each step quickly by the MS and RSEM. The other is that we could predict the bottleneck or critical situation quickly by executing proper steps forward. This can also evaluate the predefined rules and factors to obtain the better scheduling rules for the MS system at the same time.

Simulations Result

We have implemented a simulation program in Java and have run the simulations on the NetBeans IDE 4.1. We have done two types of simulations for a Least Slack (LS) time scheduling policy and our proposed MS approach. The LS time scheduling has been developed in the research, Fluctuation Smoothing Policy for Mean Cycle Time (FSMCT) (Kumar and Kumar 2001) in which the FSMCT is for re-entrant production lines. The entire class of LS policies has been proven stable in a deterministic setting (Lu and Kumar 1991) (Kumar 1994). The LS scheduling policy sets the highest priority to a wafer lot whose slack time is the smallest in the queue buffer of one machine. When the machine is going to idle, it will select the highest priority wafer lot in the queue buffer to service next. For simplifying the simulation to easily represent the scheduling methods, we have made the following assumptions:

- *Each wafer lot has the same process steps and quantity.*
- *Each photolithography stage has the same process time.*
- *There is no breakdown event in the simulations.*
- *There is unlimited capacity for non-photolithography machines.*
- *We assume the time period is very large; therefore we do not need to consider the Utilization factor in the simulations.*

Our simulations are to use two photolithography machines and 1000 wafer lots. Each wafer lot in the first simulation has 28 steps, and 5 of them are photolithography stages. While each wafer lot in second simulation has 41 steps and 9 of them are photolithography stages. For examples, in the following two simulation patterns, "0" is to represent the non-photolithography stage, "2" is to represent the photolithography stage. The wafer lot, W_2, starts to process in the factory when W_1 has passed two steps. W_3 starts when W_1 has passed three steps. We tried to simulate the arrival rate between two wafer lots as a Poisson distribution.

Simulation I:

Wafer lot pattern (28 Steps, 5 Photolithography Steps)

W_1:*0020000200000000200000200200*
W_2: *0020000200000000200000200200*
W_3: *0020000200000000200000200200*
:
W_{1000}: *0020000200000000200000200200*

Simulation II:

Wafer lot pattern (49 Steps, 9 Photolithography Steps)

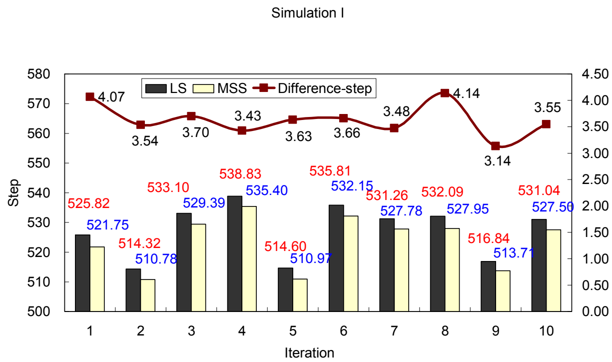
W_1:*0020000200000000200000200020020020020020 0*
W_2: *00200002000000002000002000200200200200200*
W_3: *00200002000000002000002000200200200200200*
:
W_{1000}: *00200002000000002000002000200200200200200*

We applied the LS and MS methods for these two photolithography machines to select next wafer lot in the simulations. When the wafer lot needs to wait for its limited machine, we will insert a "1" in the process pattern of the wafer lot to represent the situation. After finishing the simulations, we count the numbers ("0", "1", and "2") in the pattern of wafer lots to realize how many steps they have used. We ran ten iterations for both the simulation I and II and Figure 3 shows the simulations result.

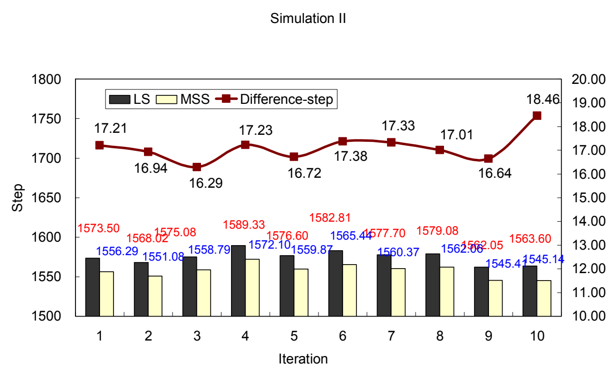
By comparing ten iterations of mean cycle time, the LS has an average 527.37 steps in simulation I and 1574.78 steps in simulation II. The MS method has an average 523.74 and 1557.66 steps in simulation I and II. The MS method is better than LS 0.69% in simulation I and 1.99% in simulation II, respectively.

Although the simulations are simplified, they reflect the real situation we have met in the factory. After applying the LS method to the above simulations and counting the required resource (RR(r_k^x, S_j), k =2, x = 1 and 2) for the photolithography machines at each step, we can realize that the load of two machines becomes unbalanced during the simulations, as well as the thrashing phenomenon that happens in the simulations. It's not difficult to extend the simulation with more machines, wafer lots, and stages. Moreover, we can use different numbers of "2" ("2", "22", or "2222") together for the task patterns to represent different process time for different photolithography stage.

We also want to point out that the more constrained photolithography stages the wafer lot has, the better the MS is than the LS method in the simulations.



(a) 5 photolithography stages



(b) 9 photolithography stages

Figure 3: Simulation results

Conclusion

To solve the issue of load balancing among photolithography machines the proposed MS system based on RSEM has been presented. The advantage of MS system is that agents could dynamically adjust the loading status to achieve the load balancing among the photolithography machines. The MS system could be extended easily. A new *Plan agent* having the same agent functions for different product can handle more different products. *Machine agent* can easily extend the system by applying a new agent to a new machine as well. Moreover, the multiagent system architecture is easy for practicing other manufacturing problems in the area with a similar constraint.

In the future work, we want to apply machine learning to set the weight of factors automatically depending on the status of the production line and the historical contracts. Some experienced managers could only set the weight of factors according to their special need manually now.

References

Akcalt, E.; Nemoto, K.; and Uzsoy, R. 2001. Cycle-time improvements for photolithography process in semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing* 14(1): 48-56.

Arisha, A., and Young, P. 2004. Intelligent Simulation-based Lot Scheduling of Photolithography Toolsets in a Wafer Fabrication Facility. In *Winter Simulation Conference*, 1935-1942.

Koestler, A. 1967. *The Ghost in the Machine*. Arkana Books, London, UK.

Kumar, P. R. 1993. Re-entrant Lines. In *Queuing Systems: Theory and Applications, Special Issue on Queuing Networks* 13(1-3): 87-110.

Kumar, P.R. 1994. Scheduling Manufacturing Systems of Re-Entrant Lines. *Stochastic Modeling and Analysis of Manufacturing Systems*, 325-360. David D. Yao (ed.), New York: Springer-Verlag.

Leitão, P.; Colombo, A. W.; and Restivo, F. J., 2005. ADACOR: A Collaborative Production Automation and Control Architecture. *IEEE Intelligent System*. 20(1):58-66.

Lou, R. C.; Shr, A. M. D.; and Hu, C. Y. 1998. Multiagent Based Multisensor Resource Management System. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems*, Victoria, BC, Canada, 1034–1039.

Lu, S. C. H.; Ramaswamy, D.; and Kumar, P. R. 1994. Efficient Scheduling Policies to Reduce Mean and Variance of Cycle-time in Semiconductor Manufacturing Plants. *IEEE Transactions on Semiconductor Manufacturing* 7(3): 374-385.

Maturana, F. P.; Staron, R. J.; and Hall, K. H. 2005. Methodologies and Tools for Intelligent Agents in Distributed Control. *IEEE Intelligent System*. 20(1):42-49.

Maˇrík, V. and McFarlane, D. 2005. Industrial Adoption of Agent-Based Technologies. *IEEE Intelligent System*. 20(1):27-35.

Miwa, T.; Nishihara, N.; and Yamamoto, K. 2005. Automated Stepper Load Balance Allocation System. *IEEE Transactions on Semiconductor Manufacturing* 18(4): 510-516.

Mönch, L.; Prause, M.; and Schmalfuss V. 2001. Simulation-Based Solution of Load-Balancing Problems in the Photolithography Area of a Semiconductor Wafer Fabrication Facility. In *Winter Simulation Conference*, 1170-1177.

Pˇechouˇcek, M.; Vokˇrínek, J.; and Beˇcváˇr, P. 2005. ExPlanTech: Multiagent Support for Manufacturing Decision Making. *IEEE Intelligent System*. 20(1):67-74.

Shr, A. M. D.; Liu, A.; and Chen, P. P. 2006a. A Load Balancing Scheduling Approach for Dedicated Machine Constraint. In *ICEIS2006, Proceedings of the 8th International Conference on Enterprise Information Systems,* Paphos, Cyprus, May 2006. Forthcoming.

Shr, A. M. D.; Liu, A.; and Chen, P. P. 2006b. A Heuristic Load Balancing Scheduling Method for Dedicated Machine Constraint. In *Proceedings of the 19th International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems (IEA/AIE'06),* Annecy, France, June 2006. Forthcoming.

Van Leeuwen E. H., and Norrie, D. 1997. Intelligent Manufacturing: Holons and Holarchies. *Manufacturing Engineer*, 76(2): 86-88.