Manufacturing Schedule of Dual-armed Cluster Tools Based on Heuristic Search

Yanfeng Geng, Kai Kang, Ji Liu, and Hong Wang, *Member, IEEE*

***Abstract*—Cluster tools are fundamental equipments in over-8-inch wafer manufacturing. Due to the tools' unique physical structure and particular semiconductor manufacturing demands, the scheduling of cluster tools is more complex than normal Job-Shop problems. This paper proposes a scheduling algorithm for dual-armed cluster tools based on Heuristic Search. We mainly examine the scheduling technique under the most complex situation including residency constraints and multi-visit requirements. A periodic idea that every wafer enters into cluster tools in the same interval is proved still feasible under this situation. To find a conflict-free schedule under a fundamental period (FP) in short time, we define an evaluation function to select the most proper operation sequence from all the possibilities, and some feedback information from a conflict is used to prevent over pruning. Experiments under the most complex situations demonstrate the feasibility and effectiveness of this algorithm. Other experiments with looser constraints show its expansibility and adaptability.**

I. Introduction

THE semiconductor manufacturing and liquid crystal display manufacturing industry increasingly use integrated equipment, including cluster tools, track equipment, and in-line equipment [1], [2]. Especially when the wafer size is increasing to 300mm, thanks to their flexible, reconfigurable and efficient characters [3], [4], cluster tools are playing a more and more important role in chip factories. A cluster tool consists of a number of processing modules (PM), a transport module (TM), and one or more cassette modules (CM). Raw materials enter the system through the CM, which interfaces with the outside environment, visit one or more PMs in a specific order, and return back to the CM they have been originated from. A single-arm or dual-arm TM is responsible for moving the wafers between modules in the cluster tool. When all the materials in a CM (commonly contains 25 wafers according to SEMI standard) are processed, the loadlock will be unloaded and replaced with another loadlock containing a new set of raw units.

Scheduling a cluster tool involves specifying a sequence of TM actions and their times to move wafers between PMs in order to satisfy a number of constraints and objectives [5]. The scheduling problem in cluster tools can be quite complicated due to stringent timing and throughput requirements. One important constraint is the residency time, which requires the wafer to leave the PM after it has been processed in a limit time. If the wafer delay at some process steps exceeds the residency time, it is quite possible that the wafer surface deteriorate and should be scrapped. Another specific problem is the multi-visit requirement, which means that, according to some specific processing recipes, a wafer may visit the same PM more than one time. In such circumstance, we should carefully arrange the TM transporting sequence to avoid exceeding the residency time and falling into the deadlock.

Our objective is to find an effective scheduling algorithm for dual-armed cluster tools with residency constraints and multi-visit requirements. Besides, to satisfy the various requirements in the actual wafer fabricating, this algorithm should also be feasible and quick enough to solve problems under time-free and single-visited situations. In this paper, we propose a quick algorithm based on heuristic search to fulfill the above requirements.

The remainder of this paper is organized as follows. Section two describes the model and provides definitions for the terms used throughout the paper. Section three discusses the related works. Section four presents the proposed scheduling algorithm. Section five provides the results of our experiments. Section six makes a conclusion of our algorithm.

II. Model and Definitions

Fig. 1 shows two kinds of typical cluster tools. The first one consists one dual-arm robot, while the second one contain one dual-arm robot to move wafers between PMs and temporary plate (TP) and another single-arm robot to move wafers between TP and CM. Since the actions of single-arm robots are simple and spend little time, we focus our attention on arranging actions of the dual-arm robot.

It seems that cluster tool scheduling problem is just one kind of Job-Shop scheduling problems, however, it possesses its own particularity. The first uniqueness is the transportation. All the movements of wafers are performed by only one dual-arm robot, so its operation sequences should be carefully designed to both fully use the two arms and avoid overburden .The second unique character is that most cluster tools consist parallel PMs. Parallel PMs have the same function (e.g., cooling, heating), so wafers can choose each of them to finish the processing step. The third

This work was supported in part by the National 863 Program of China under Grant 2006AA03A141

Yanfeng Geng, Ji liu and Hong Wang are with Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, 110016(e-mail: gengyf@sia.cn; liuji@sia.cn; wang@sia.cn).

Kai Kang is with Shenyang Microcyber Automation Technology Ltd. Shenyang, 110016 (e-mail: kang.kaixiao@microcyber.cn)

978-1-4244-1706-3/08/$25.00 ©2008 IEEE.

one is the residency constraint, which greatly enhances the difficulties to decide the action sequences and action time of the dual-arm robot. The last one is the multi-visit requirement, which dramatically disturbs the normal sequences and can easily cause the deadlock.

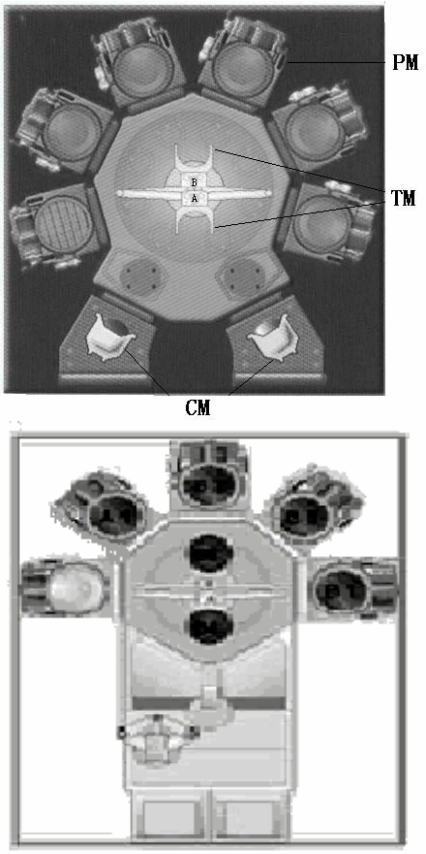

Fig. 1.Two typical cluster tools

Based on the above particular constraints and normal constraints of Job-Shop problems, we set up a mathematical model for cluster tool scheduling problem as following.

$$\min \max_{1\le k\le M} \{\max_{1\le i\le N} t_{ik}\} \tag{1}$$

s.t.

$$t_{ik} - p_{ik} \ge t_{ih} + T_{trans}, \quad for\ i = 1,2,...,N\ and\ h,k = 1,2,...,M,\ \ h < k \tag{2}$$

$$t_{jk} - t_{ik} \ge p_{jk}, \quad for\ i,j = 1,2,...,N, i < j\ and\ k = 1,2,...,M \tag{3}$$

$$0 \le p_{ik} - T_{proc_k} \le T_{res_k} \tag{4}$$

$$T_{trans} = T_{pick} + T_{place} + T_{hold} \tag{5}$$

$$T_{proc_k}, T_{pick}, T_{place} > 0, \quad T_{res_k}, T_{hold}, t_{ik} \ge 0, \quad for\ i = 1,2,...,N\ and\ k = 1,2,...,M \tag{6}$$

One of the common metrics of cluster tool performance is the throughput. The final goal of a schedule is to find a feasible operation sequence that costs the least time to process N wafers, as (1) portrays, which in other words, brings the largest throughput in the same processing time.

TABLE I
SOME TERM DEFINITIONS

Symbol	Definition
M	Total number of steps of a process
N	Number of wafers to be processed
V_i	Visit i
t_{ik}	Time when i wafer finishes being processed in the k visit
p_{ik}	Time i wafer spends for the k visit
T_{trans}	Robot action time to move one wafer from one PM to another
T_{pick}	Pick time
T_{place}	Place time
T_{hold}	Hold time
T_{proc_k}	Processing time of the k visit
T_{res_k}	Residency constraint of the k visit
$T_{\max}$	Maximum of T_{proc_k}
FP	Interval period
BN	Basic number to increase

Constraints (2) and (3) restrict the basic processing sequence determined by the recipe and guarantee that one PM can only process one wafer at the same time.

Constraint (4) shows that a wafer must remain in one of the PMs for at least T_{proc_k} to be completely processed and after being processed it should leave the PM in less than T_{res_k} .If a wafer leaves one of the PM before T_{proc_k} , it will be premature, and if it stays more than $T_{proc_k} + T_{res_k}$,it violates the residency constraint.

A TM transfers wafers from one module to another. It performs three kinds of actions: *pick* wafers from a source module, *place* wafers into a destination module, and *move* in order to reach a source module or destination module. A dual-armed robot has two blades pointing in opposite directions. Comparing with single-arm robot, it has two advantages. The first one is that one of the arms can be used as a temporary buffer. For example, when a wafer is picked up from the source PM but the destination PM is occupied by another wafer, the wafer can temporarily stay in one arm until the destination PM is ready. In this situation, the other arm can still transport wafers just as a single-arm robot. The time from picking to placing a wafer is called T_{hold} so the time to transport a wafer from source to destination PM, called T_{trans} , is defined as (5) . The second advantage is that a dual-armed robot can use a *swap* operation method. *Swap* is such an action that the robot unloads a processed wafer from a module into an empty robot arm and immediately loads the next wafer from another arm into the module. This

advantage reduces the moving times comparing to the single-arm robot, and allows for more wafers processing simultaneously in the cluster tools.

III. RELATED WORKS

The previous work on modeling and optimization of cluster tools has been mostly focused on single-wafer and single-visit cluster tools.

Perkinson et al. [6], and Wood [7] assumed that the TM has only one arm, and has no residency constraints. They analyzed what the ideal fundamental period (FP) can be, and since residency constraints were not considered, finding efficient algorithms for the optimal periodic schedule was possible. Perkinson divided the operations of the system into two different regions. The first one, the transport-bound region, happens when the TM is always busy with moving the wafers. During each period, the TM should perform exactly $(M+1)$ *pick* actions and $(M+1)$ *place* actions; therefore, if we assume $T_{pick} = T_{place} = T$,the ideal FP would be $2T(M+1)$ The second, called process-bound region, happens when the TM has some idle time. The step needing maximum processing time becomes the throughput bottleneck, and should be kept busy all the time, so the ideal FP would be $T_{\max} + 4T$. Fig. 2 shows the two kinds of scheduling method with Ideal FP .

Vankatesh et al. [8] proposed an optimal solution for a dual-armed cluster tool, but without addressing any form of residency constraints in their model. With the unique *swap* action, the FP in dual-armed cluster tools becomes $T_{\max} + 2T$ in the process-bound region.

Rostami et al. [9] examined heuristic algorithms for finding an operation sequence for a dual-armed cluster tool with the wafer delay constraints. Ja-Hee Kim et al. [10] and Tae-Eog Lee [11], [12] constructed a Petri net model for time-constrained dual-armed cluster tools.

Perfect jobs as they did in modeling and solving the scheduling problem [13]-[15], we are not aware of any work on cluster tools that addressed the multi-visit requirements, which, however, is quite common in practicality applications. For example, in a Track machine, wafers need be heated before and after coating, and if the machine has only one heating PM, it is inevitable for a wafer to visit heating PM twice. In this paper, we examine an algorithm for finding an effective schedule for dual-armed cluster tools with residency constraints and multi-visit requirements.

IV. ALGORITHM DESCRIPTION

A. Basic idea

We assume that the number of the wafers to be processed is usually much larger than the number of the processing steps, namely $N >> M$. In the steady state that all processing steps are being performed, we can find that there is still a constant interval between any neighboring finished wafers, even under the multi-visit situation. Therefore, we can still make use of Perkinson's periodic scheduling model.

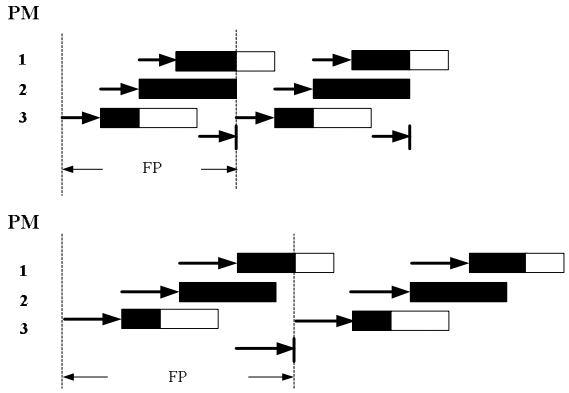

Fig. 2.Two kinds of scheduling with Ideal FP

In the single-visit situation without residency constraint, the ideal FP can be described as following:

$$FP = \max\{T_{\max} + 2T_{trans}, 2(M+1)T_{trans}\}$$
$$T_{\max} = \max\{T_{proc_k}\}, k = 1, 2, ..., M \quad (7)$$

where $T_{\max}$ is the longest processing time of all the modules. But if the cluster tool contains parallel modules, for example, if there are n parallel modules that can process the k step, the T_{proc_k} should be adjusted as (8) [16].

$$T_{eff} = \frac{T_{proc_k} + 2T_{trans}}{n} - 2T_{trans} \quad (8)$$

However, when adding residency constraint, wafers can not enter into the cluster tools with the ideal FP interval. We can conclude from Fig. 2 that every wafer will remain in the PM for $(T_{inv} - T_{proc_{ik}} - 2T_{trans})$, and if this period of time is longer than T_{res_k} , the wafer is over processed. Besides, the multi-visit condition also breaks the normal action sequence. If wafers keep entering into the cluster tool every FP time, there will be too many wafers in the tool simultaneously. The multi-visited PM is busy when a wafer previously processed reaches again, in this situation, the wafer in the PM can not be picked and the wafer in the robot can not be placed, which is called deadlock. When a deadlock happens, the residency constraints are broken.

Obviously, the actual FP is larger than the ideal one, and to examine an effective schedule that generates largest throughput, we should find the smallest interval. To satisfy this basic demand, we gradually increase FP by a fixed number. Each time the FP is increased, we test all the possible robot action sequences. If the sequence is feasible to properly process all the wafers, we stop increasing FP and the current FP is the least one; if we can not find a feasible sequence under the current FP , we should keep increasing the FP until we find a proper one. The basic thought can be described as following.

1. Calculate Ideal FP
2. While not Done
   2.1 Increase FP by BN , schedule
   2.2 If no conflict, Done

2.3 If conflict exist, continue

3. End while

B. Complexity analysis

There are tremendous amount of possible scheduling sequences and choices under a FP . Firstly, every step of the processing recipe has residency constraint T_{res_k} , so when a wafer is processed in a PM, it can be picked up at any time between 0 and T_{res_k} , which is not an easy choice. If it is picked up too early, one of the arms will be occupied when the destination PM is not free; if it is picked up too late, the total time for processing one wafer will increase. Secondly, when it comes to a conflict, as Fig.3 shows, there are many ways to free the conflict. For example, if two wafers are completed simultaneously, the robot can pick one wafer up immediately and let the other stay in the PM, or pick both of them and let one of them stay in one arm temporally, or reschedule to let one of the wafers enter into the current PM later by making them spending more time in the previous steps. Each possibility is feasible if it can server to complete the whole processing under current FP. Thirdly, if there are parallel modules to serve for next processing step, we should make a decision which one is the best to choose. Therefore, the searching space for a feasible schedule is so large that will cost unacceptable amount of time. To find a feasible and effective schedule in short time, we examine a technique based on heuristic search.

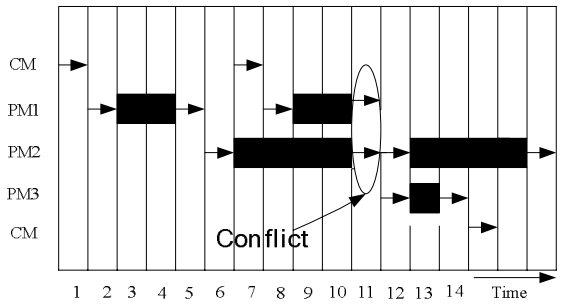

Fig. 3.Conflict: two wafers need to be picked up at the same time

C. Heuristic Search

Heuristic Search is a well established, fundamental field of research in Artificial Intelligence. Many hard problems can be modeled as pathfinding in a state-space graph. An intelligent search will be guided by heuristics so as to solve problems quickly.

Heuristic Search starts from an initial state in the state-space, expands the search tree to new states based on the heuristic rules and information, and finally reaches the objective state. The heuristic rule is depicted in the evaluation function:

$$f(n) = g(n) + h(n) \tag{9}$$

where $f(n)$ is the evaluation value of n node; $g(n)$ is the actual value from initial node to n node; $h(n)$ is an estimating value from n node to final node. The heuristic information is contained in $h(n)$.

In our scheduling problem, we define

$$h(i) = p_{ik} + \sum_{k<j\le m} T_{proc_{ij}} + (N-i)T_{inv} \tag{10}$$

$$g(i) = t_{ik} \tag{11}$$

where $h(i)$ estimates the rest processing time after the k step of i wafer. The least value of $h(i)$ implies that we tend to pick up the wafer from the module as soon as it is processed. And when it comes to a conflict, we choose the wafer that needs less time to process the rest steps.

Under the strict constraints of both residency and multi-visit, the final FP is often much more than the ideal one, which indicates that we might repeat the same searching process many times. To reduce the searching time, rather than examine all expanded nodes, we only examine the most valuable ones according the heuristic information and neglect others. We tend to choose the nodes that mostly save the time under the current FP , and if the deadlock or overtime happens, we study feedback information of the wafers which are responsible for the conflict. For example, if i wafer exceeds the residency time while the robot is busy moving other wafers, the information about the i wafer draws our attention. According to the information about the conflict, usually the time that a wafer exceeds the residency constraint, we rechoose another alternative of previous nodes which is as near to the current one as possible. If we can find a node that can free the current conflict, then we reexpand the braches after the node and keeping search for the final schedule. If we can not find a proper one from the previous nodes to solve the conflict, it implies that the current FP is too small, so we increase it by BN and repeat the same procedure until a feasible one is found. The value of BN depends on the step processing time. The larger BN reduces searching time but renders more processing time, while the smaller one is just the opposite.

The final Heuristic Search algorithm is depicted as following:

1) Calculate initial FP
2) Create initial node, Open Table and Close Table. Put the initial node into Open table.
3) If Open Table is empty , $FP+ = BN$, Go to 2)
4) Choose the node with the least $f(i)$, named node n , put it into Close Table
5) If the node n is final node, return
6) If node n can be expanded, calculate $f(m_i)$ of the new expanded m_i nodes and put the node with least $f(m_i)$ into Open Table. If the node has existed, replace it.
7) If node n can not be expanded, check the corresponding nodes for the conflict in the Close Table, find the nearest one that can free the conflict by choose its another alternative, and put it back into Open Table.
8) Go to 3)

D. Algorithm analysis

The algorithm we design is mainly used in the situation

that the number of wafers to be processed is much larger than the processing steps (if $N < M$, the periodic idea is not useful for a good result). To reduce searching time, our searching starts from an ideal FP , which firstly excludes a large amount of impossible scheduling sequences. Then under each FP, rather than testing all the possibilities, the complexity of which is

$$O(\prod_{0<i<N,0<k<M}(T_{res_{ik}}+1)*N*M)$$

, we only choose the most possible one according to heuristic evaluation function, and the complexity becomes $O(N*M)$. Therefore the time spent to find a feasible schedule greatly reduces. Additionally, to explore an effective algorithm which can cut down the total processing time as much as possible, we distill the feedback information from the conflict, locate the causing step from previous scheduling track, regulate the time and continue to search from that step. A good regulation, on one hand, takes more possibly feasible scheduling sequences into consider, which are neglected by the evaluation function; on the other hand, it avoids retrying other unnecessary scheduling steps that soon cause another conflict. In short, Combining the evaluation function and the strategy of utilizing the feedback information, this algorithm tends to find an effective schedule in very short time.

V. EXPERIMENT RESULT

We test this algorithm in a Track cluster tool. The Track tool contains four PMs: the first PM called COT to coat material on the wafer, the second called CP to cool wafers, and the remained two called HP to heat wafers.

We configure a typical processing recipe, as table 2 shows. There are five steps in total, and all wafers visits HP twice. Each wafer should leave PM after it is processed in no more than 1 second, namely $T_{res_{ik}} = 1s$ .We also assume: $T_{place} = T_{pick} = 1s$, and neglect the time that the dual-armed robot costs to move from one PM to another.

6 wafers are to be processed according to the above processing recipe. Using our algorithm, the final processing sequence is showed in Fig.4. Wafers enter into the cluster tool at an interval of 24s, and the total processing time is 195s, which is, though not the best, a good result. There is no deadlock and none wafer exceeds the residency constraint, so it is a feasible and effective schedule. This experiment is conducted in a computer with 2.8GHz Intel CPU and 512 RAM, and it costs only 165ms to generate this result.

More experiments are conducted under different situations. Firstly, we increase the number of wafers to 50, 80 and 100. we find that FP keeps 24s at each circumstance, and the total searching time increases only a little. From the result we can conclude that most of the time is cost to find a feasible schedule for the first n wafers, where n is determined by the total steps and each step-time of the processing recipe. Wafers after the n are just repeating the former scheduling sequence. Hence, if the number of wafers to be processed is very large, we can just search the scheduling sequences for the first n ones, which must be large enough to reach the steady state, and repeat the scheduling sequence at steady state for the rest wafers. Different recipes lead to different n , and with large amount of practice, we find that $n = 3M$ is enough to cover each wafer's scheduling sequences. The scheduling sequences of the middle M wafers delegate the steady-state situation. Secondly, different kinds of processing recipes are configured to test the robust and adaptability of the algorithm. TABLE 3 shows the searching results for 6 wafers under 4 different recipes. We change the position of bottleneck step, change the multi-visit PM and remove the multi-visit condition, and under each situation, we can find a feasible schedule.

TABLE 2
PROCESSING RECIPE

Processing Step	Processing Time(s)
HP	20
CP	15
COT	10
HP	20

In addition, we try to remove the residency constraint to examine the expansibility of the algorithm. Under this simpler precondition, when there is no multi-visit in the recipe, we can find the ideal scheduling result as Perkinson did; when multi-visit is required in the recipe, an effective result that costs relatively little time can still be generated with this algorithm. With the recipe as TABLE 2 shows, the final FP equals to 19ms without the residency constraints.

TABLE 3
SCHEDULING UNDER DIFFERENT PROCESSING RECIPES

Processing Steps and Step time (s)	FP(s)	Searching Time(ms)
HP(20)->CP(15)->COT(10)->HP(20)	24	165
HP(10)->CP(15)->COT(30)->HP(18)	32	158
CP(15)->COT(10)->HP(20)	17	136
HP(20)->CP(15)->COT(10)->CP(20)	29	219

VI. CONCLUSION

In this paper, we propose a scheduling method for dual-armed cluster tools with residency constraints and multi-visit requirements. To satisfy the demands of both the short generating time and effectiveness of the schedule, we solve this problem based on Heuristic Search and some improving techniques. The results of abundant experiments show that the scheduling technique can find both feasible and relatively effective schedule costing very little time. Besides, when it comes to looser conditions, such as without time constraint or single-visit, effective schedules can still be generated. Therefore, this algorithm serves as an effective method to scheduling dual-armed cluster tools under most conditions.

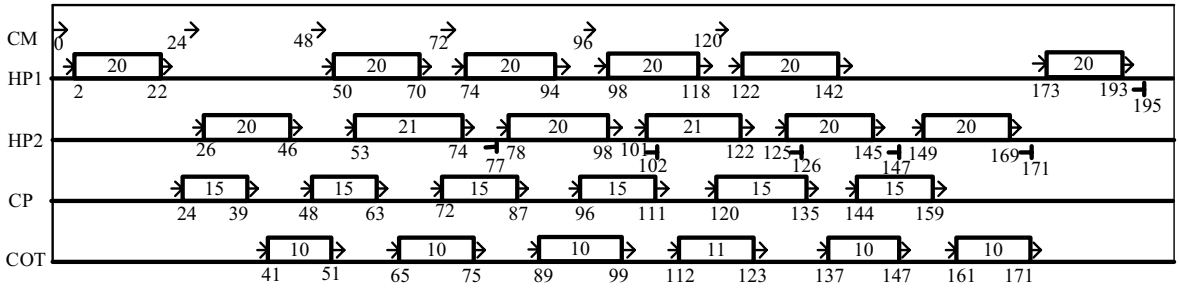

Fig. 4. Scheduling diagram for the testing experiment

REFERENCES

[1] Jin-Hwan Lee and Tae-Eog Lee, " A supervisory Equipment Control Application Model for Integrated Semiconductor Manufacturing Equipment," in *IEEE Robotics & Automation Magazine,* March 2004, pp. 41-58

[2] S. C. Wood and K. C. Saraswat, "Modeling the performance of clusterbased fabs," in *Proc. IEEE/CHMT Int. Symp. Electronics Manufacturing Technology*, 1991, pp. 8–14.

[3] M. E. Bader, R. P. Hall, and G. Strasser, "Integrated processing equipment," *Solid State Technol.*, vol. 33, pp. 149–154, May 1990.

[4] P. Burggraaf, "Coping with the high cost of wafer fabs," *Semicond. Int.*, vol. 38, pp. 45–50, March 1995.

[5] S. Rostami, B. Hamidzadeh, and D. Camporese, "An optimal periodic scheduler for dual-arm robots in cluster tools with residency constraints," *IEEE Trans. Robot. Automat.*, vol. 17, pp. 609–618, Oct.2001.

[6] T. L. Perkinson, P. K. McLarty, R. S. Gyurcsik, and R. K. Cavin III,"Single-wafer cluster tool performance: An analysis of throughput,"*IEEE Trans. Semicond. Manufact.*, vol. 7, pp. 369–373, Aug. 1994.

[7] S. C. Wood, "Simple performance models for integrated processing tools," *IEEE Trans. Semicond. Manufact.*, vol. 9, pp. 320–328, Aug. 1996.

[8] S. Venkatesh, R. Davenport, P. Foxhoven, and J. Nulman, "A steadystate throughput analysis of cluster tools: Dual-blade versus single-blade robots," *IEEE Trans. Semiconduct. Manufact.*, vol. 10, pp. 418–424, Nov. 1997.

[9] S. Rostami, B. Hamidzadeh, and D. Camporese," Optimal scheduling techniques for cluster tools with process-module and transport-module residency constraints," *Proceedings of the 2001 IEEE Conference on Robotics and Automation,* 2001

[10] Ja-Hee Kim, T.-E. Lee, H.-Y. Lee and D.-B. Park,"Scheduling Anslysis of Time-Constrained Dual-Armed Cluster Tools," *IEEE Transactions on Semiconductor Manufacturing*, vol. 16, NO.3. August 2003

[11] T.-E. Lee and Y.-H. Shin, "Scheduling time-constrained cyclic manufacturing systems: Linear programming and directed graph models," in *Proc. 4th Asian-Pacific Conf. Industrial Engineering and Management Systems*, 2002, pp. 1–5..

[12] Y.-H. Shin, T.-E. Lee, J.-H. Kim, and H.-Y. Lee, "Modeling and implementating a real-time scheduler for dual-armed cluster tools," *Comput.in Industry*, vol. 45, no. 1, pp. 13–27, 2001.

[13] R. S. Srinivasan, "Modeling and performance analysis of cluster tools using Petri nets," *IEEE Thans. Semiconduct. Manufact.*, vol. 11, pp. 394–403, Aug. 1998.

[14] R. A. Hendrickson, "Optimizing cluster tool throughput," *Solid State Technol.*, pp. 217–222, July 1997.

[15] H. Chen, C. Chu, and J. M. Proth, "Cyclic scheduling of a hoist with time window constraints," *IEEE Trans. Robot. Automat.*, vol. 14, pp. 144–152, Feb. 1998.

[16] T. L. Perkinson, R. S. Gyurcsik, and P. K. McLarty, "Single-wafer cluster too performance: An analysis of the effects of redundant chambers and revisitation sequences on throughput," *IEEE Trans. Semicond. Manufact.*,vol. 9, pp. 384–400, Aug. 1996