Journal of Southeast University (English Edition) Vol. 29, No. 2, pp. 187 - 193 June 2013 ISSN 1003—7985

Scheduling method for single-arm cluster tools of wafer fabrications with residency and continuous reentrancy

Zhou Binghai Wang Zhu Chen Jia

(School of Mechanical Engineering, Tongji University, Shanghai 201804, China)

Abstract: In order to enhance the utilization of single-arm cluster tools and optimize the scheduling problems of dynamic reaching wafers with residency time and continuous reentrancy constraints, a structural heuristic scheduling algorithm is presented. A nonlinear programming scheduling model is built on the basis of bounding the scheduling problem domain. A feasible path search scheduling method of single-arm robotic operations is put forward with the objective of minimal makespan. Finally, simulation experiments are designed to analyze the scheduling algorithms. Results indicate that the proposed algorithm is feasible and valid to solve the scheduling problems of multiple wafer types and single-arm clusters with the conflicts and deadlocks generated by residency time and continuous reentrancy constraints.

Key words: cluster tools; scheduling; residency time; continuous reentrancy

doi: 10.3969/j. issn. 1003 - 7985. 2013. 02. 014

With the rapid development of the semiconductor manufacturing technology, especially the diameter of wafers expanding to 300 mm, cluster tools are adopted widely in the modern semiconductor industry. Cluster tools have diverse scheduling requirements such as complex wafer flow patterns, wafer residency time constraints, and resource constraints, etc. It is difficult to solve their scheduling problems. How to improve the scheduling performance is of great significance to reduce cost and shorten the cycle time of semiconductor wafer fabrications.

Venkatesh et al. [1] studied how to schedule robotic tasks of single-arm cluster tools, but their methods are based on the assumption of wafer processing without residency time constraints. Currently, there is some related literature on scheduling problems of single-arm cluster tools with residency time constraints. Lee et al. [2-3] studied the scheduling problem of the minimum period under the steady state and established an algorithm to search for the minimum period, but the calculation is too complex. Rostami et al. [4-5] studied the schedulability problem of cluster tools and considered residency time constraints. They presented a linear programming-based method to find an optimal periodic schedule with a buffer module. However, the method only considered the scheduling problem of a single wafer product type. Yoon and Lee[6] established a scheduling model with residency time constraints and put forward a kind of online scheduling algorithm with different product types. The literature mentioned above has not considered continuous reentrant processing requirements.

Perkinson et al. [7] presented an analysis model of the effect of redundant processing chambers and chamber reentrant process sequences on steady-state throughput. Lee et al. [8] used Petri net models to develop deadlock avoidance conditions and built a mixed integer programming model. However, the mathematical method is not suitable for solving large-scale scheduling problems with reentrant wafer flows. Zuberek et al. [9-10] developed timed Petri nets to model and analyze scheduling problems of cluster tools with chamber reentrancy. Nevertheless, only the static scheduling problem of robot operations is considered. Jung et al. [11] proposed a mixed integer programming model to minimize the cycle time of timed Petri net models of cluster tools with various scheduling requirements. Wu et al. [12-13] developed resource-oriented Petri net models with colors and time introduced to describe the operations of cluster tools. Kim et al. [14] determined the cycle time of a cluster tool with the timed event graph model. Wu et al. [15] developed a formal Petri net model to address the real-time operational problem with residency time constraints. These methods are applicable to various reentrant processing types, but none addresses residency time constraints.

The methods mentioned above cannot fully adapt to the practical applications when residency time and reentrancy constraints are separately considered to study the scheduling problem of single-arm cluster tools. At present, there are only a few works on scheduling problems of single-arm cluster tools with residence time and reentrancy constraints. In this paper, the scheduling problem of single-arm cluster tools is explored with residency time constraints, continuous reentrancy constraints and diverse wa-

Received 2012-10-29.

Biography: Zhou Binghai (1965—), male, doctor, professor, bhzhou@tongji. edu. cn.

Foundation item: The National Natural Science Foundation of China (No. 71071115, 61273035).

Citation: Zhou Binghai, Wang Zhu, Chen Jia. Scheduling method for single-arm cluster tools of wafer fabrications with residency and continuous reentrancy[J]. Journal of Southeast University (English Edition), 2013, 29(2): 187 - 193. [doi: 10. 3969/j. issn. 1003 - 7985. 2013. 02. 014]

fer product types.

1 Problem Formulations

A single-arm cluster tool consists of several process modules (PMs), a transport module (TM), and loadlocks (LLs). The logic chart is shown in Fig. 1. The PMs execute wafer processes. The LLs are used for storing the unprocessed and processed wafers. The TM unloads a wafer from one LL to the PM, loads it back to the other LL, and moves the wafer from one PM to another PM.

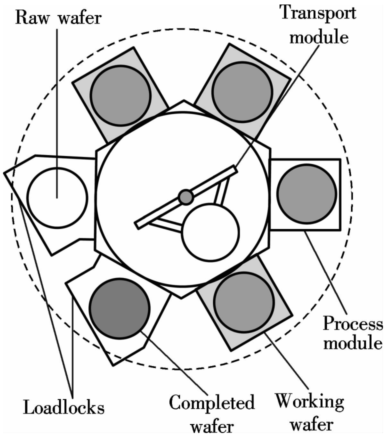

Fig. 1 Single-arm cluster tool logic chart

To effectively formulate a scheduling problem domain, the following assumptions are given:

1) The TM is a single-arm robot, which can pick, place or transport only one wafer at a time; the time of loading, unloading and transporting a wafer among different PMs and LLs can be variable.

2) Each PM only processes one wafer at a time.

3) There exists the phenomenon of consecutive reentrant processing between two adjacent PMs, i. e., the same wafer returning to the same two consecutive PMs.

4) There are residency time constraints for wafers in PMs. The time interval of residency in PMs is (a, b), and $(0, \infty)$ in the LLs. If the actual residency time of a wafer exceeds the upper limit of the time interval, the wafers should be considered as being of bad quality.

5) Residency time and processing time for each wafer within each PM can be different.

6) A wafer lot is scheduled at the moment of arriving at the LL.

7) There is no buffer among PMs. A cluster tool of three PMs is considered in this paper.

When researching on the scheduling problem of cluster tools with residency time and reentrancy constraints, there are some certain requirements that must be satisfied. Starting time $t^b_{\sigma(i),j,h}$ and finishing time $t^c_{\sigma(i),j,h}$ for processing a wafer in a sequence must obey the following inequalities:

$$t^b_{\sigma(i_1),j_1} \geqslant t^b_{\sigma(i_2),j_2} + t_{\mathrm{H}} + t_{\mathrm{U}} + t_{\mathrm{L}} \tag{1}$$

$$t^c_{\sigma(i_1),j_1} \geqslant t^c_{\sigma(i_2),j_2} + t_{\mathrm{H}} + t_{\mathrm{U}} + t_{\mathrm{L}} \tag{2}$$

$$t^c_{\sigma(i_1),j_1} \geqslant t^b_{\sigma(i_2),j_2} + t_{\mathrm{H}} + t_{\mathrm{U}} + t_{\mathrm{L}} \tag{3}$$

where n is the total number of wafers in a lot; $\sigma_{(k)}$ is the k-th wafer to be processed, $k = 1, 2, \dots, n$; σ is the order of wafers to be processed, $\sigma = (\sigma_{(1)}, \sigma_{(2)}, \dots, \sigma_{(k)}, \dots, \sigma_{(n)})$; t_{H}, t_{U}, and t_{L} are defined as the transport time between PMs and LLs, the unloading time of a wafer from a PM to the LL, and the loading time of a wafer from a PM to the LL, respectively; $t^b_{\sigma(i),j,h}$ represents the time point that the i-th wafer enters the j-th PM for the h-th processing; $t^c_{\sigma(i),j,h}$ refers to the time point that the i-th wafer leaves the j-th PM for the h-th processing.

According to assumption 2), a PM only processes one wafer at a time such that the PM must be free before a new wafer is inserted. Namely,

$$t^b_{\sigma(i+1),j} \geqslant t^c_{\sigma(i),j} + t_{\mathrm{H}} + t_{\mathrm{U}} + t_{\mathrm{L}} \tag{4}$$

As mentioned above, this paper addresses the scheduling with continuous reentrancy. Two adjacent PMs which process a wafer more than one time should satisfy the following requirements:

$$t^b_{\sigma(i),j,h_2} \geqslant t^c_{\sigma(i),j,h_1} + 2t_{\mathrm{H}} + 2t_{\mathrm{U}} + t_{\mathrm{L}} \tag{5}$$

$$t^b_{\sigma(i),j,h_2} \geqslant t^c_{\sigma(i),j+1,h_1} + t_{\mathrm{H}} + t_{\mathrm{U}} + t_{\mathrm{L}} \tag{6}$$

The actual time that a wafer resides in a PM should be at least equal to processing time $T^h_{\sigma(i),j}$, and at most the sum of processing time and residency time $t^h_{\sigma(i),j,\mathrm{U}}$. The following relationships must be satisfied:

$$t^h_{\sigma(i),j} \geqslant T^h_{\sigma(i),j} \tag{7}$$

$$t^h_{\sigma(i),j} \leqslant T^h_{\sigma(i),j} + t^h_{\sigma(i),j,\mathrm{U}} \tag{8}$$

To make full use of the equipment, the processing operation begins once a wafer is loaded in the PM. The details are described as

$$t^c_{\sigma(i),j,h} = t^b_{\sigma(i),j,h} + t^h_{\sigma(i),j} \tag{9}$$

$$t^b_{\sigma(i),j_2} = t^c_{\sigma(i),j_1} + t_{\mathrm{H}} + t_{\mathrm{U}} \tag{10}$$

$$w^h_{j(i)} \leqslant t^h_{\sigma(i),j} - t_{\mathrm{U}} - t_{\mathrm{L}} \tag{11}$$

where $w^h_{j(i)}$ represents the TM waiting time at the j-th PM for the h-th processing of the i-th wafer; $t^h_{\sigma(i),j}$ is the actual residency time of the i-th wafer in the j-th PM for the h-th processing.

According to assumption 3), two PMs processing each wafer with continuous reentrancy must satisfy the following equation:

$$t^b_{\sigma(i),j+1,h} = t^c_{\sigma(i),j,h} + t_{\mathrm{H}} + t_{\mathrm{U}} \tag{12}$$

Let π be the wafer processing routing and $\pi = \{C_1^{x_1}, C_2^{x_2}, \dots, C_n^{x_n}\}$, where $C_i^{x_i}$ is the processing times in the x_i-th

PM, and subscript i is the i-th processing step. So the routing of wafers is $\pi = [1_1^1, 1_2^2, 1_3^3, 2_4^1, 2_5^2]$ in this paper, i. e. LL→PM1→PM2→PM3→PM1→PM2→LL. The wafer reentrant flow pattern is shown in Fig. 2.

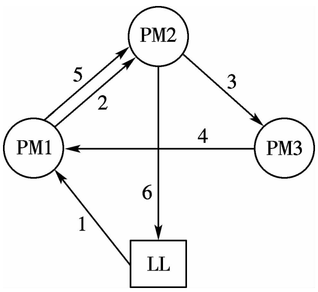

Fig. 2 Wafer reentrant flow pattern

Let L_i be the i-th TM operation path, and Z_i be the state of each PM under periodic scheduling when path L_i is adopted, $Z_i = \{L_1, L_2, \ldots, L_n\}$. The i-th PM has two states O and Θ. O and Θ denote that the PM is empty and occupied, respectively. A set of five TM operational paths under the steady period is $Z_i = \{L_1, \Theta, \ldots, L_5\}$. They are $L_1 = (R_1 \to R_2 \to R_3 \to R_4 \to R_5 \to R_6)$, $L_2 = (R_1 \to R_6 \to R_2 \to R_4 \to R_3 \to R_5)$, $L_3 = (R_6 \to R_1 \to R_2 \to R_4 \to R_3 \to R_5)$, $L_4 = (R_1 \to R_6 \to R_2 \to R_3 \to R_4 \to R_5)$, $L_5 = (R_1 \to R_2 \to R_3 \to R_4 \to R_6 \to R_5)$.

The scheduling objective function is to minimize the system makespan under path L_i, i. e., to minimize the $C^i_{\max}$ when all the constraints are satisfied,

$$\min C^i_{\max} \tag{13}$$

Consequently, the scheduling problem contains the objective function (13) and the constraints (1) to (12).

2 Scheduling Algorithm

In order to solve the scheduling problem about various kinds of wafers with residency time and continuous reentrancy constraints, a scheduling model is established with three PMs in single-arm cluster tools. A feasible path search scheduling method is put forward for robotic operations. The routing of wafers is $\pi = [1_1^1, 1_2^2, 1_3^3, 2_4^1, 2_5^2]$. We assume that the wafer processing in PM1 and PM2 has to be done twice through continuous reentrancy. Namely, the scheduling method is used to find out all the feasible path sets of scheduling TM satisfying all residency time and reentrancy constraints under different Z_i states.

The core of the proposed scheduling algorithm is to adopt the pull strategy[1]. Due to many kinds of deadlocks caused by reentrant processes, all the possible TM paths satisfying reentrancy constraints must be listed so that feasible paths can be found out. The task of processing a wafer lot both in the initial state and the intermediate state must satisfy residency time and reentrancy constraints. The makespan of the feasible scheduling solution is calculated. Finally, the minimal makespan solution is determined.

The proposed path search scheduling method includes the following two parts: The first part is to find out each scheduling path L_i which can satisfy all residency time and reentrancy constraints during each wafer's operations in turn; the second part is to calculate the corresponding makespan and obtain an optimal scheduling path. The algorithm procedure is described in details as follows:

Step 1 Determine whether L_2 is a feasible sequence path or not. If the path meets the constraints, calculate the system makespan, otherwise determine the next path.

1) Let $i = 1$; $j = 0$.

2) Let $i = i + 1$, $k = 1$, then calculate $t^1_{\sigma(1),2}$, $t^1_{\sigma(2),1}$, $t_{\sigma(1),3}$, $t^1_{\sigma(2),2}$, $t^2_{\sigma(1),1}$ as

$$t^1_{\sigma(1),2} = t_U + 3t_H + t_L - T^1_{\sigma(1),2} \tag{14}$$

$$t^1_{\sigma(2),1} = t_U + 3t_H + t_L + w^1_{2(1)} - T^1_{\sigma(2),1} \tag{15}$$

$$t_{\sigma(1),3} = t_U + 3t_H + t_L + w^1_{1(2)} - T_{\sigma(1),3} \tag{16}$$

$$t^1_{\sigma(2),2} = t_U + 3t_H + t_L + w_{3(1)} - T^1_{\sigma(2),2} \tag{17}$$

$$t^2_{\sigma(1),1} = t_U + 3t_H + t_L + w^1_{2(2)} - T^2_{\sigma(1),1} \tag{18}$$

3) If $t^1_{\sigma(1),2} > t^1_{\sigma(1),2,U}$ or $t^1_{\sigma(2),1} > t^1_{\sigma(2),1,U}$ or $t_{\sigma(1),3} > t_{\sigma(1),3,U}$ or $t^1_{\sigma(2),2} > t^1_{\sigma(2),2,U}$ or $t^2_{\sigma(1),1} > t^2_{\sigma(1),1,U}$, go to Step 2.

4) Let $k = k + 1$, then calculate $t^2_{\sigma(k),2}$, $t^1_{\sigma(k+2),1}$, $t_{\sigma(k+1),3}$, $t^1_{\sigma(k+2),2}$, $t^2_{\sigma(k+1),2}$, $t^2_{\sigma(k),1}$ as

$$t^2_{\sigma(k),2} = t_U + 3t_H + t_L - T^2_{\sigma(k),2} \tag{19}$$

$$t^1_{\sigma(k+2),1} = t_U + 3t_H + t_L + w^2_{\sigma(k)} - T^1_{\sigma(k+2),1} \tag{20}$$

$$t_{\sigma(k+1),3} = 4t_U + 8t_H + 4t_L + w^2_{1(k)} + w^2_{2(k)} + w^1_{1(k+2)} - T_{\sigma(k+1),3} \tag{21}$$

$$t^1_{\sigma(k+2),2} = t_U + 3t_H + t_L + w_{3(k+1)} - T^1_{\sigma(k+2),2} \tag{22}$$

$$t^2_{\sigma(k+1),2} = t_U + 3t_H + t_L - T^2_{\sigma(k+1),2} \tag{23}$$

$$t^2_{\sigma(k),1} = t_U + 3t_H + t_L + w^1_{2(k+1)} - T^2_{\sigma(k),1} \tag{24}$$

5) If $t^2_{\sigma(k),2} > t^2_{\sigma(k),2,U}$ or $t^1_{\sigma(k+2),1} > t^1_{\sigma(k+2),1,U}$ or $t_{\sigma(k+1),3} > t_{\sigma(k+1),3,U}$ or $t^1_{\sigma(k+2),2} > t^1_{\sigma(k+2),2,U}$ or $t^2_{\sigma(k+1),2} > t^2_{\sigma(k+1),2,U}$ or $t^2_{\sigma(k),1} > t^2_{\sigma(k),1,U}$, go to Step 2.

6) If $t^2_{\sigma(k),2} \leqslant t^2_{\sigma(k),2,U}$, $t^1_{\sigma(k+2),1} \leqslant t^1_{\sigma(k+2),1,U}$, $t_{\sigma(k+1),3} \leqslant t_{\sigma(k+1),3,U}$, $t^1_{\sigma(k+2),2} \leqslant t^1_{\sigma(k+2),2,U}$, $t^2_{\sigma(k+1),2} \leqslant t^2_{\sigma(k+1),2,U}$, $t^2_{\sigma(k),1} \leqslant t^2_{\sigma(k),1,U}$ and $k \leqslant n-2$, then go back to 4).

7) If $k > n-2$, use Eqs. (25), (26) and (27) to calculate $t_{\sigma(n),3}$, $t^2_{\sigma(n-1),2}$, $t^2_{\sigma(n),1}$ as

$$t_{\sigma(n),3} = t_U + 3t_H + t_L + w^2_{1(n-1)} - T_{\sigma(n),3} \tag{25}$$

$$t^2_{\sigma(n-1),2} = t_U + 3t_H + t_L + w_{3(n)} - T^2_{\sigma(n-1),2} \tag{26}$$

$$t^2_{\sigma(n),1} = t_U + 3t_H + t_L + w^2_{2(n-1)} - T^2_{\sigma(n),1} \tag{27}$$

8) If $t_{\sigma(n),3} > t_{\sigma(n),3,U}$ or $t^2_{\sigma(n-1),2} > t^2_{\sigma(n-1),2,U}$ or $t^2_{\sigma(n),1} >$

$t^2_{\sigma(n),1,U}$, go to Step 2.

9) Use Eq. (28) to calculate $C^i_{\max}$ as

$$C^i_{\max} = \sum_{k=1}^{n-2} [w^2_{2(k)} + w^1_{1(k+2)} + w_{3(k+1)} + w^1_{2(k+2)} + w^2_{2(k+1)}] + (12n-3)t_H + 6n(t_U + t_L) + T^1_{\sigma(1),1} + w^1_{2(1)} + w^1_{1(2)} + w_{3(1)} + w^1_{2(2)} + w^2_{1(1)} + w_{3(n)} + w^2_{2(n-1)} + w^1_{1(n)} + T^2_{\sigma(n),2} \quad (28)$$

Step 2 Determine whether L_3 is a feasible sequence path or not. If the path meets the constraints, calculate the system makespan, otherwise determine the next path.

1) Let $i=i+1$ and $k=0$.

2) Let $k=k+1$, then calculate $t_{\sigma(k+1),3}$, $t^1_{\sigma(k+2),2}$, $t^2_{\sigma(k+1),1}$ as

$$t_{\sigma(k+1),3} = 4t_U + 6t_H + 4t_L + w^2_{1(k)} + T^2_{\sigma(k),2} + T^1_{\sigma(k+2),1} - T_{\sigma(k+1),3} \quad (29)$$

$$t^1_{\sigma(k+2),2} = t_U + 3t_H + t_L + w_{3(k+1)} - T^1_{\sigma(k+2),2} \quad (30)$$

$$t^2_{\sigma(k+1),1} = t_U + 3t_H + t_L + w_{3(k+1)} + w^1_{2(k+2)} - T^2_{\sigma(k+1),1} \quad (31)$$

3) If $t_{\sigma(k+1),3} > t_{\sigma(k+1),3,U}$ or $t^1_{\sigma(k+2),2} > t^1_{\sigma(k+2),2,U}$ or $t^2_{\sigma(k+1),1} > t^2_{\sigma(k+1),1,U}$, go to step 3.

4) If $t_{\sigma(k+1),3} \leqslant t_{\sigma(k+1),3,U}$, $t^1_{\sigma(k+2),2} \leqslant t^1_{\sigma(k+2),2,U}$, $t^2_{\sigma(k+1),1} \leqslant t^2_{\sigma(k+1),1,U}$ and $k \leqslant n-1$, then go back to 2).

5) If $k > n-1$, use Eq. (32) to calculate $C^i_{\max}$ as

$$C^i_{\max} = \sum_{k=1}^{n-2} [T^2_{\sigma(2),2} + T^1_{\sigma(k+2),1} + w_{3(k+1)} + w^1_{2(k+2)} + w^2_{1(k+1)}] + (9n+3)t_H + 6n(t_U + t_L) + T^1_{\sigma(1),1} + w^1_{2(1)} + w^1_{1(2)} + w^1_{3(1)} + w^1_{2(2)} + w^2_{1(1)} + w_{3(n)} + w^2_{2(n-1)} + w^2_{1(n)} + T^2_{\sigma(n),2} \quad (32)$$

Step 3 Determine whether L_4 is a feasible sequence path. If the path meets the constraints, calculate the system makespan, otherwise determine the next path.

1) Let $i=i+1$ and $k=0$.

2) Let $k=k+1$, then calculate $t^2_{\sigma(k),2}$, $t^1_{\sigma(k+1),1}$ as

$$t^2_{\sigma(k),2} = t_U + 3t_H + t_L - T^2_{\sigma(k),2} \quad (33)$$

$$t^1_{\sigma(k+1),1} = t_U + 3t_H + t_L + w^2_{2(k)} - T^1_{\sigma(k+1),1} \quad (34)$$

3) If $t^2_{\sigma(k),2} > t^2_{\sigma(k),2,U}$ or $t^1_{\sigma(k+1),1} > t^1_{\sigma(k+1),1,U}$, go to Step 4.

4) If $t^2_{\sigma(k),2} \leqslant t^2_{\sigma(k),2,U}$, $t^1_{\sigma(k+1),1} \leqslant t^1_{\sigma(k+1),1,U}$ and $k \leqslant n-2$, then go back to 2).

5) If $k > n-2$, use Eqs. (35) and (36) to calculate $t^2_{\sigma(n-1),2}$, $t^1_{\sigma(n),1}$ as

$$t^2_{\sigma(n-1),2} = t_U + 3t_H + t_L - T^2_{\sigma(n-1),2} \quad (35)$$

$$t^1_{\sigma(n),1} = t_U + 3t_H + t_L + w^2_{2(n)} - T^1_{\sigma(n),1} \quad (36)$$

6) If $t^2_{\sigma(n-1),2} > t^2_{\sigma(n-1),2,U}$ or $t^1_{\sigma(n),1} > t^1_{\sigma(n),1,U}$, go to Step 4.

7) Use Eq. (37) to calculate $C^i_{\max}$ as

$$C^i_{\max} = \sum_{k=1}^{n-2} [T^1_{\sigma(k+1),2} + T_{\sigma(k+1),3} + T_{\sigma(k+1),1} + w^2_{2(1)} + w^1_{1(k+1)}] + (9n-3)t_H + 6n(t_U + t_L) + T^1_{\sigma(1),1} + T^1_{\sigma(1),2} + T_{\sigma(1),3} + T^2_{\sigma(1),1} + w^2_{2(n-1)} + w^1_{1(n)} + T^1_{\sigma(n),2} + T_{\sigma(n),3} + T^2_{\sigma(n),1} + T^2_{\sigma(n),2} \quad (37)$$

Step 4 Determine whether L_5 is a feasible sequence path or not. If the path meets the constraints, calculate the system makespan, otherwise determine the next path.

1) Let $i=i+1$ and $k=0$.

2) Let $k=k+1$, then calculate $t^1_{\sigma(2k-1),2}$, $t^1_{\sigma(2k),1}$, $t_{\sigma(2k-1),3}$, $t^1_{\sigma(2k),2}$, $t^2_{\sigma(2k-1),1}$, $t_{\sigma(2k),3}$, $t^2_{\sigma(2k-1),2}$ as

$$t^1_{\sigma(2k-1),2} = t_U + 3t_H + t_L - T^1_{\sigma(k),2} \quad (38)$$

$$t^1_{\sigma(2k),1} = t_U + 3t_H + t_L + w^1_{2(k)} - T^1_{\sigma(k+1),1} \quad (39)$$

$$t_{\sigma(2k-1),3} = t_U + 3t_H + t_L + w^1_{1(k+1)} - T_{\sigma(k),3} \quad (40)$$

$$t^1_{\sigma(2k),2} = t_U + 3t_H + t_L + w_{3(k)} - T^1_{\sigma(k+1),2} \quad (41)$$

$$t^2_{\sigma(2k-1),1} = t_U + 3t_H + t_L + w^1_{2(k+1)} - T^2_{\sigma(k),1} \quad (42)$$

$$t_{\sigma(2k),3} = t_U + 3t_H + t_L + w^2_{1(k)} - T_{\sigma(k+1),3} \quad (43)$$

$$t^2_{\sigma(2k-1),2} = t_U + 3t_H + t_L + w_{3(k+1)} - T^2_{\sigma(k),2} \quad (44)$$

3) If $t^1_{\sigma(2k-1),2} > t^1_{\sigma(2k-1),2,U}$ or $t^1_{\sigma(2k),1} > t^1_{\sigma(2k),1,U}$ or $t_{\sigma(2k-1),3} > t_{\sigma(2k-1),3,U}$ or $t^1_{\sigma(2k),2} > t^1_{\sigma(2k),2,U}$ or $t^2_{\sigma(2k-1),1} > t^2_{\sigma(2k-1),1,U}$ or $t_{\sigma(2k),3} > t_{\sigma(2k),3,U}$ or $t^2_{\sigma(2k-1),2} > t^2_{\sigma(2k-1),2,U}$, go to Step 5.

4) If $t^1_{\sigma(2k-1),2} \leqslant t^1_{\sigma(2k-1),2,U}$, $t^1_{\sigma(2k),1} \leqslant t^1_{\sigma(2k),1,U}$, $t_{\sigma(2k-1),3} \leqslant t_{\sigma(2k-1),3,U}$, $t^1_{\sigma(2k),2} \leqslant t^1_{\sigma(2k),2,U}$, $t^2_{\sigma(2k-1),1} \leqslant t^2_{\sigma(2k-1),1,U}$, $t_{\sigma(2k),3} \leqslant t_{\sigma(2k),3,U}$, $t^2_{\sigma(2k-1),2} \leqslant t^2_{\sigma(2k-1),2,U}$ and $k \leqslant [n/2]$, then go back to 2).

5) If $k \leqslant [n/2]$, use Eq. (45) to calculate $C^i_{\max}$ as

$$C^i_{\max} = \sum_{k=1}^{n-1} \frac{1}{2}[T^1_{\sigma(k),1} + w^1_{2(k)} + w^1_{1(k+1)} + w_{3(k)} + w^1_{2(k)} + w^2_{1(k)} + w_{3(k+1)} + w^2_{2(k)} + T^2_{\sigma(k+1),1} + T^2_{\sigma(k+1),2}] + (21n-9)\frac{t_H}{2} + 6n(t_U + t_L) + T^1_{\sigma(1),1} + T^1_{\sigma(n),2} + T_{\sigma(n),3} + T^2_{\sigma(n),1} + T^2_{\sigma(n),2} \quad (45)$$

Step 5 Calculate the makespan by scheduling path L_1, and find out the best scheduling path in all the possible sequence paths:

1) Let $i=i+1$, then use Eq. (46) to calculate $C^i_{\max}$ as

$$C^i_{\max} = \sum_{k=1}^{n} [6(t_H + t_U + t_L) + T^1_{\sigma(k),1} + T^1_{\sigma(k),2} + T_{\sigma(k),3} + T^2_{\sigma(k),1} + T^2_{\sigma(k),2}] \quad (46)$$

2) Calculate min $C^i_{\max}$, and find out the optimal scheduling solution.

Step 6 The algorithm ends.

3 Example Analysis

The wafer flow pattern is $\pi = \{1^1_1, 1^2_2, 1^3_3, 2^1_4, 2^2_5\}$, $T^1_{\sigma(i),1}$

$=70$, $T^1_{\sigma(i),2}=75$, $T_{\sigma(i),3}=85$, $T^2_{\sigma(i),1}=70$, $T^1_{\sigma(i),2}=75$, $t_H=2$, $t_U=t_L=1$, $i=25$, $t^1_{\sigma(i),1,U}=40$, $t^1_{\sigma(i),2,U}=45$, $t_{\sigma(i),3,U}=45$, $t^2_{\sigma(i),1,U}=40$, $t^2_{\sigma(i),2,U}=45$. After running the proposed algorithm, we have $C^4_{\max}=8\ 343$, $C^5_{\max}=7\ 099$, $C^1_{\max}=10\ 075$, and then choose the best feasible path L_5. The optimal scheduling for wafer continuous reentrant processing is shown in Fig. 3.

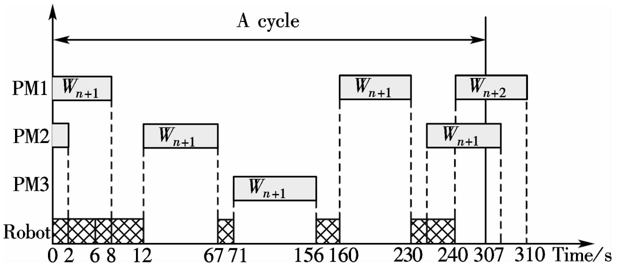

Fig. 3 Gantt chart for L_5

Fig. 3 shows that the reentrancy of the third process module makes other two process modules' utilization reduced, but the phenomena will not cause deadlocks. Meanwhile, path L_5 can make residency time that is spent in the third process module very short, only 1 s, and also make other process modules have no residency. The example indicates that we can find out an optimal robot scheduling path with the proposed algorithm.

4 Simulation and Analyses

To effectively evaluate the system performance of the proposed approach, some variables are defined as follows:

$R_s=\dfrac{C_{\max}-C_{\min}}{C_{\max}}\times 100\%$ is defined as the improvement rate of results obtained by the proposed algorithm compared with results obtained by the general algorithm which schedules one piece of wafer each time. The more the R_s value, the less the makespan value.

$R_x=\dfrac{P_C-P_A}{P_C}\times 100\%$ is defined as the similarity rate of results obtained by the proposed algorithm compared with the results obtained by the algorithm proposed in Ref. [6]. The smaller the R_x value, the more similar to the optimal P; that is, the better performance the proposed algorithm has.

$B_R=\dfrac{T_{\max}}{(N-1)(t_L+t_U+t_H)}$ denotes the relationship between the maximum processing time and the transport time. The greater the B_R value, the more idle time of the TM. The less the B_R value, the more the idle time of the PMs. $T_{\max}$ denotes the maximum processing time in the cluster tool; N is the total number of processing steps.

We program the heuristic scheduling algorithm in Visual C + + language and run it on a computer. Based on the experimental results, the following analyses are given.

4.1 Analysis of general running time of the algorithm

Let $T^h_{\sigma(i),j}\sim N(70,10)$; $t^h_{\sigma(i),j,U}\sim N(40,5)$; $t_U=t_L=1$, $t_H=2$. The simulation experiments are carried out on a personal computer with a 160 GB hard disk, 1 GB DDR3 memory and 1.6 GHz Intel CoreTMi3 processor 330M. Fig. 4 shows the running time of the proposed algorithm under the number of wafers from 10 to 100.

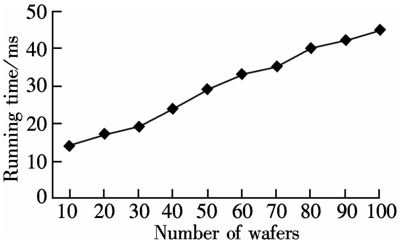

Fig. 4 Running time of the algorithm vs. number of wafers

Fig. 4 indicates that the running time of the algorithm increases along with the increase in the number of wafers. But it is obvious that the running time of our algorithm is very short. The running time is 45 ms when the number of wafers is 100. The proposed algorithm is suitable for carrying out a real-time scheduling of the wafers.

4.2 Algorithm performance under impact of B_R

Let $T^h_{\sigma(i),j}\sim N(70,10)$; $t_U=t_L=1$, the number of wafers in a lot is $i=25$. The residency time obeys a normal distribution. The expected value of residency time is 40 and the standard deviation is 0, 5, 10, 15 and 20. When B_R changes from 1 to 8, the tendencies of R_s are shown in Fig. 5.

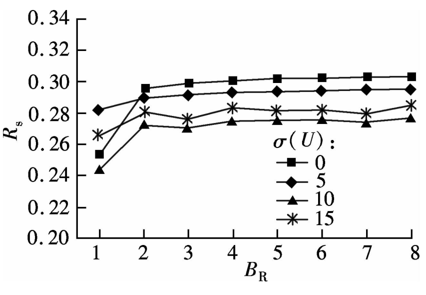

Fig. 5 B_R vs. R_s

Fig. 5 indicates that the makespan calculated by the proposed algorithm becomes smaller, and R_s keeps improving. When the standard deviation of the residency time increases, the general variation tendency of R_s is very small. The standard deviation of the residency time has little influence on the performance of the proposed algorithm. The curve of R_s becomes smooth after B_R is greater than 2, namely, the performance becomes relatively steady. In practice, the transport time is always very small relative to the maximum processing time, and the proposed algorithm is feasible and valid to schedule cluster tools with residency time and continuous reentrancy constraints.

4.3 Impact of T on algorithm performance

Let $t^h_{\sigma(i),j,\mathrm{U}} \sim N(40, 5)$; $t_{\mathrm{U}} = t_{\mathrm{L}} = 1$; $t_{\mathrm{H}} = 2$. T obeys a normal distribution and the expected value is 40, 50, 60, 70, 80. The standard deviation is 5. The value of R_x varies with different numbers of wafers and the average processing time, and the results are shown in Fig. 6.

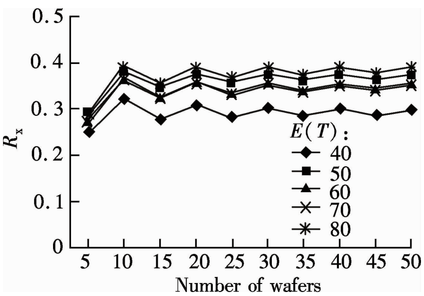

Fig. 6 R_x vs. number of wafers and processing time AVG

Fig. 6 shows that under the constraints of residence, R_x changes within a small range from 25% to 40% when the average processing time $E(T)$ changes. And R_x becomes small and regressive along with the increasing number of wafers. It is an expected and perfect result. When the wafer number i is more than 20, the R_x value tends to stabilize.

4.4 Impact of size of wafer batch on algorithm performance

Let $T \sim N(70, 10)$; $t^h_{\sigma(i),j,\mathrm{U}} \sim N(40, 5)$; $t_{\mathrm{U}} = t_{\mathrm{L}} = 1$; $t_{\mathrm{H}} = 2$. Ten different samples of wafers are tested to obtain the values of R_s and R_x. According to the definitions of R_s and R_x, the algorithm performance can be verified by various R_s and R_x.

As shown in Fig. 7, R_s increases with the enlarging number of wafers. Although floating exists, R_s still ranges from 20% to 30%. R_x declines but tends to be stable around 35%. And the FP is closer to the ideal FP. In practice, the number of wafers equals the result obtained by using the proposed algorithm, so our algorithm performs well in application.

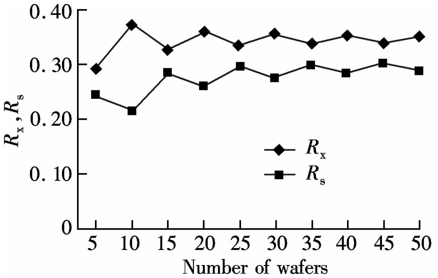

Fig. 7 R_x/R_s vs. number of wafers

5 Conclusion

1) The proposed algorithm can effectively solve the scheduling problem of multiple wafer types and single-arm clusters with the conflicts and deadlocks generated by residency time and continuous reentrancy constraints.

2) The feasibility and availability of the developed heuristic scheduling algorithm are verified in Visual C + + language. Due to the short running time of the algorithm, the proposed algorithm can solve a real-time scheduling problem of the wafers.

3) Compared with the swap policy algorithm, the experimental results indicate that the proposed algorithm has good performance.

References

[1] Venkatesh S, Davenport R, Foxhoven P, et al. A steady-state throughput analysis of cluster tools: dual-blade versus single-blade robots[J]. *IEEE Transactions on Semiconductor Manufacturing*, 1997, **10**(4): 418 - 424.

[2] Lee T E, Park S H. An extended event graph with negative places and tokens for timed window constraints[J]. *IEEE Transactions on Automation Science and Engineering*, 2005, **2**(4): 319 - 332.

[3] Kim J H, Lee T E, Park D B, et al. Scheduling analysis of time-constrained dual-armed cluster tools [J]. *IEEE Transactions on Semiconductor Manufacturing*, 2003, **16**(3): 521 - 534.

[4] Rostami S, Hamidzadeh B. An optimal periodic scheduler for dual-arm robots in cluster tools with residency constraints[J]. *IEEE Transactions on Robotics and Automation*, 2001, **17**(5): 609 - 618.

[5] Rostami S, Hamidzadeh B. An optimal residency-aware scheduling technique for cluster tools with buffer module [J]. *IEEE Transactions on Semiconductor Manufacturing*, 2004, **17**(1): 68 - 73.

[6] Yoon H J, Lee D Y. On-line scheduling of integrated single wafer processing tools with temporal constraints[J]. *IEEE Transactions on Semiconductor Manufacturing*, 2006, **18**(3): 390 - 398.

[7] Perkinson T L, Gyurcsik R S, McLarty P K. Single-wafer cluster tool performance: an analysis of the effects of redundant chambers and revisitation sequences on throughput[J]. *IEEE Transactions on Semiconductor Manufacturing*, 1996, **9**(3): 384 - 400.

[8] Lee H Y, Lee T E. Scheduling single-arm cluster tools with reentrant wafer flows [J]. *IEEE Transactions on Semiconductor Manufacturing*, 2006, **19**(2): 226 - 240.

[9] Zuberek W M. Cluster tools with chamber revisiting-modeling and analysis using timed Petri nets[J]. *IEEE Transactions on Semiconductor Manufacturing*, 2004, **17**(3): 333 - 344.

[10] Zuberek W M. Petri net modeling and performance analysis of cluster tools with chamber revisiting[C]//*The 8th International Conference on Emerging Technologies and Factory Automation*. Antibes-Juan les Pins, France, 2001, **2**: 105 - 112.

[11] Jung C Y, Lee T E. An efficient mixed integer program-

ming model based on timed Petri nets for diverse complex cluster tool scheduling problems[J]. *IEEE Transactions on Semiconductor Manufacturing*, 2012, **25**(2): 186 − 199.

[12] Wu N Q, Chu F, Chu C B, et al. Petri net-based scheduling of single-arm cluster tools with reentrant atomic layer deposition processes[J]. *IEEE Transactions on Automation Science and Engineering*, 2011, **8**(1): 42 − 55.

[13] Wu N Q, Chu F, Chu C B, et al. Petri net-based cycle time analysis of dual-arm cluster tools with wafer revisiting and swapping strategy[C]//2011 *IEEE International Conference on Robotics and Automation*. Shanghai, China, 2011: 5499 − 5504.

[14] Kim D K, Jung C Y, Lee T E, et al. Cyclic scheduling of cluster tools with non-identical chamber access times [C]//*Proceedings of the* 2011 *Winter Simulation Conference*. Phoenix, AZ, USA, 2011: 2068 − 2079.

[15] Wu N Q, Zhou M C. Schedulability analysis and optimal scheduling of dual-armed cluster tools with residency time constraint and activity time variation[J]. *IEEE Transactions on Automation Science and Engineering*, 2012, **9**(1): 203 − 209.

带驻留与连续重入的单臂集束型晶圆制造设备调度方法

周炳海　王　翥　陈　佳

（同济大学机械与能源工程学院，上海 201804）

摘要：为了提升单臂集束型设备的利用率，优化动态到达晶圆驻留与连续重入的调度问题，提出了一种结构式启发式调度算法. 在界定调度问题域的基础上，建立了非线性规划的调度模型，并以动态到达晶圆的最短完工时间为调度目标，构造了基于搜索可行机械手搬运路径的调度算法. 最后，设计了仿真实验，并对调度算法进行了实验分析. 结果表明，所提出的算法对于解决多种晶圆类型的调度问题以及单臂集束型晶圆制造设备在加工过程中由于驻留约束限制和连续重入而产生的冲突和死锁的问题是可行而有效的.

关键词：集束型设备；调度；驻留时间；连续重入

中图分类号：TP391