

考虑多晶圆流的集束型设备群调度方法

周炳海, 黎明

(同济大学 机械与能源工程学院, 上海 201804)

摘 要: 为有效解决集束型设备群多品种晶圆调度问题, 构建考虑多晶圆流集束型设备群调度算法. 统筹考虑多晶圆流、驻留和资源约束等特征, 并以系统 Makespan 最小为目标, 建立数学模型. 集束型设备群调度分解为外层晶圆序列决策和内层机械手调度两个阶段. 采用蚁群寻觅结合双向搜索策略, 将蚁群算法融合到晶圆时间约束集搜索算法迭代过程中. 在此基础上, 提出了 AS&TC(ant systems and time constraints)调度算法. 对该调度算法进行仿真实验分析, 结果表明该算法的有效性.

关 键 词: 集束型设备群; 多晶圆流; 驻留约束; 时间约束集; 启发式算法

中图分类号: TP 29 文献标志码: A 文章编号: 1005 - 3026(2016)05 - 0697 - 05

Scheduling Method for Multi-cluster Tools with Diverse Wafer Flow Patterns

ZHOU Bing-hai, LI Ming

(School of Mechanical Engineering, Tongji University, Shanghai 201804, China. Corresponding author: ZHOU Bing-hai, professor, E-mail: bhzhou@tongji.edu.cn)

Abstract : To deal with scheduling problems of multi-cluster tools (MCTs) with different wafer types effectively, a scheduling algorithm considering diverse wafer flow patterns was put forward. Considered the wafer flow patterns, resource and residency constraints, mathematical programming models were set up with an objective of minimizing the Makespan. The scheduling problem of the single-armed MCTs was divided into two stages: determine the processing sequence of wafer lots and schedule the operating sequence of the robot arms. Applying ant colony search and bidirectional search strategies, ant system algorithm was introduced into the process of dynamic and adaptive time constraint set iterations. A new scheduling algorithm was built based on the ant systems and time (AS & TC). Simulation experiments results indicated that the algorithm is feasible.

Key words : multi-cluster tools(MCTs); diverse wafer flow patterns; residency constraints; time constraint set; heuristic algorithm

随着半导体制造进入 450 mm 晶圆时代, 集束型设备群(multi-cluster tools, MCTs)被普遍使用. MCTs 调度属于典型 NP 难问题, 具有多晶圆流、驻留及缓冲区资源约束等特殊特性, 使得传统单集束型设备(single-cluster tool, SCT)调度方法^[1-4]无法满足 MCTs 调度要求, 因此 MCTs 调度问题亟待解决.

MCTs 调度研究尚处于初始阶段, 目前大多数文献集中于单种晶圆稳态模式下的周期性调度研究. Yi 等^[5]将 MCTs 调度问题分解为多个 SCT

调度问题建模分析, 并求解出单种晶圆无驻留调度条件下的基本周期下界, 但其忽略机械手搬运时间. Chan 等^[6-7]将机械手搬运时间假定为固定常数, 对两种不同结构的 MCTs 进行研究, 以最小循环周期为目标, 提出了基于资源的分析建模方法. 石箫铭等^[8]考虑实际加工存在的驻留约束, 构造了顺序搜索启发式算法, 但其对象仅是双集束型设备. 上述文献均未考虑实际调度中普遍存在的多晶圆流现象.

多晶圆流是指同一个晶圆加工批次中具有多

条工艺路径。多晶圆流的研究对提高 MCTs 设备利用率和满足顾客产品多样化需求具有重大意义。

目前大多数学者对单晶圆 MCTs 调度进行了研究,对多晶圆流的多品种调度问题鲜有研究。仅 Liu 等^[9]在单臂 MCTs 中同时考虑晶圆驻留约束和多晶圆流,提出了相应的算法。但其假设晶圆的加工序列确定,未对初始晶圆进入顺序进行考虑。本文在考虑多晶圆流、驻留约束及晶圆的加工序列不确定的基础上,进一步对初始晶圆进入 MCTs 顺序进行研究,提出了 AS&TC(ant systems and time constraints)调度算法。

1 问题描述

典型多晶圆流 MCTs 如图 1 所示。晶圆按照其晶圆流从输入卡闸模块 CM_a 进入系统,依次经过各加工模块,连接相邻集束型设备的缓冲模块,最终加工完成后存放在输出卡闸模块 CM_b 。晶圆在不同模块间的转移通过搬运模块完成。图中 2 条折线分别代表 2 条不同的晶圆流,可分别用晶圆依次经过加工模块的路径坐标集 X_w 来表示。

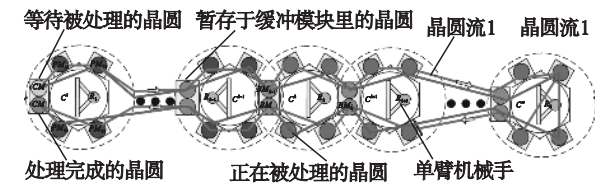


图 1 集束型设备群示意图

Fig. 1 An illustrative chart for the multi-flow pattern in single arm multi-cluster tools

C^h 表示 MCTs 中的第 h 个集束型设备, R_h 表示 C^h 的搬运模块, PM_{hj} 表示第 h 个集束型设备中的第 j 个加工模块, BM_{hj} 表示连接 C^h 和 C^{h+1} 的缓冲模块, 晶圆流路径坐标集 $X_w = \{x_1, \dots, x_i, \dots, x_{\varphi(w)}\}$, $x_i = (h, j)$ 表示晶圆第 i 步经过的处理模块序号, $\varphi(w)$ 表示晶圆 w 总加工步骤数。

问题假定: 1) 加工模块同时只能加工 1 片晶圆; 2) 搬运模块采用单臂机械手, 每次仅搬运 1 片晶圆, 且装卸时间相同; 3) 晶圆在缓冲模块驻留时间无限; 4) 晶圆加工完成后, 在加工模块内有驻留时间上限, 超过该上限, 成为次品; 5) 晶圆批具有多晶圆流, 且晶圆加工序列不确定。

据假定 1) 第 $w+1$ 片晶圆进入 PM_{hj} 的时间 $S_{hj}(w+1)$ 与第 w 片晶圆离开 PM_{hj} 的时间 $L_{hj}(w)$ 之差必定超过机械手搬运时间 t_M , 即:

$$S_{hj}(w+1) - L_{hj}(w) \geq t_M. \quad (1)$$

同理, 对缓冲模块:

$$SB_{hj}(w+1) - LB_{hj}(w) \geq t_M. \quad (2)$$

据假定 2) 晶圆在任何一个加工模块的进入时间或离开时间的差值必定超过 t_M , 即:

$$S_{Kj+1}(w) - S_{hj}(w) \geq t_M, \quad (3)$$

$$L_{Kj+1}(w) - L_{hj}(w) \geq t_M, \quad (4)$$

$$S_{Kj+1}(w) - L_{hj}(w) \geq t_M. \quad (5)$$

同理在任一缓冲模块中应满足:

$$SB_{Kj+1}(w) - SB_{hj}(w) \geq t_M, \quad (6)$$

$$LB_{Kj+1}(w) - LB_{hj}(w) \geq t_M, \quad (7)$$

$$SB_{Kj+1}(w) - LB_{hj}(w) \geq t_M. \quad (8)$$

假定 4) 晶圆在 PM_{hj} 中的实际停留时间应满足:

$$P_{hj}(w) \leq L_{hj}(w) - S_{hj}(w) \leq P_{hj}(w) + U_{hj}(w). \quad (9)$$

式中: $P_{hj}(w)$ 表示晶圆的加工时间; $U_{hj}(w)$ 表示晶圆加工完成后允许的最大驻留时间。

由假定 5) 知晶圆具有多种晶圆流模式, 所以晶圆转移须满足:

$$S_{x_{i+1}} = L_{x_i} + t_M, \quad (10)$$

$$S_{x_i} = LB_{hj} + t_M(x_{i-1} \text{ in } C^{h-1} \text{ or } C^{h+1}), \quad (11)$$

$$SB_{hj} = L_{x_i} + t_M(x_{i+1} \text{ in } C^{h-1} \text{ or } C^{h+1}). \quad (12)$$

批量为 n 的晶圆批, 晶圆序号 $\{J_1, J_2, \dots, J_n\}$ 进入 CM_a 后, 需按照顺序依次进入系统加工, 即晶圆进入首个加工模块的时间需满足:

$$S_{x_1}(q_1) \leq S_{x_1}(q_2) \leq \dots \leq S_{x_1}(q_n). \quad (13)$$

式中 q_n 为第 n 片进入系统的晶圆序号。

根据文献 [10] 对时间约束集的概念作如下定义:

$T_{R_h}(w)$ 表示对晶圆 w 调度时, 机械手 R_h 的可用时间区间集, 包含 $\varphi t(w, h)$ 个区间, 即: $T_{R_h}(w) = \{t_{R_h,1}(w), t_{R_h,2}(w), \dots, t_{R_h,\varphi(w,h)}(w)\}$; $T_{PM_{x_i}}(w)$ 表示 PM_{x_i} 可用的时间区间集, 包含 $\varphi(w, x_i)$ 个时间区间, 即: $T_{PM_{x_i}}(w) = \{t_{PM_{x_i},1}(w), t_{PM_{x_i},2}(w), \dots, t_{PM_{x_i},\varphi(w,x_i)}(w)\}$; $T_{PB_{hj}}(w)$ 表示 BM_{hj} 可用的时间区间集, 包含 $\varphi(w, h, j)$ 个区间, 即: $T_{PB_{hj}}(w) = \{t_{PB_{hj},1}(w), t_{PB_{hj},2}(w), \dots, t_{PB_{hj},\varphi(w,h,j)}(w)\}$ 。

机械手搬运必须在某个可行时间区间完成, 即:

$$\exists q \in \{1, 2, \dots, \varphi(w, h)\} : [S_{x_i}(w) - t_M, S_{x_i}(w)] \subseteq t_{R_h,q}(w). \quad (14)$$

其中 $t_{R_h,q}(w)$ 表示 $T_{R_h}(w)$ 的第 q 个可行的时间区间 $[S_{x_i}(w) - t_M, S_{x_i}(w)]$ 表示晶圆 w 从模块 $PM_{x_{i-1}}$ 搬运到模块 PM_{x_i} 所占用的机械手的时间区间。

加工模块须满足

$$\exists k \in \{1, 2, \dots, \varphi(w, h)\} : [S_{x_i}(w), L_{x_i}(w)] \subseteq t_{PM_{x_i},k}(w). \quad (15)$$

其中 $t_{PM_{x_i},k}(w)$ 表示 $T_{PM_{x_i}}(w)$ 的第 k 个可行的时间区间 $[S_{x_i}(w), L_{x_i}(w)]$ 表示晶圆 w 实际占

用模块 PM_{x_i} 的时间区间。

同理,对缓冲模块有

$$\exists p \in \{1, 2, \dots, \varphi(b, w, j)\} [SB_{hj}(w), LB_{hj}(w)] \subseteq t_{PB_{hj}p}(w). \quad (16)$$

调度目标是使系统的 Makespan 最小,即:

$$\min L_{x_n}(q_n). \quad (17)$$

本文调度问题是以式(17)为目标函数,式(1)~式(16)为约束条件的非线性规划问题。

2 算法构建

为阐明本文的算法,现作如下定义。

定义 1 晶圆流差异化因子 DFP(difference of flow pattern), $DFP_{ij} = \varphi(i) + \varphi(j) - 2\varphi_s(i, j)$; $\varphi(i)$, $\varphi(j)$ 为晶圆 J_i 和 J_j 各自的需要经过的加工模块数目, $\varphi_s(i, j)$ 表示晶圆 J_i 和 J_j 共享加工模块(即两条路径都需要经过的加工腔)的数目。

定理 1 相邻晶圆 DFP 越大,晶圆加工对资源的冲突越小,调度可获得驻留时间越短。

证明 假设晶圆 $A \setminus B \setminus C$ 三片晶圆,加工路径坐标集分别为 x_a, x_b, x_c , $\varphi(b) = \varphi(c)$, $DFP_{ab} \geq DFP_{ac}$. 则 $DFP_{ab} = \varphi(a) + \varphi(b) - 2\varphi_s(a, b)$, $DFP_{ac} = \varphi(a) + \varphi(c) - 2\varphi_s(a, c)$.

因为 $DFP_{ab} \geq DFP_{ac}$, 则 $\varphi_s(a, b) < \varphi_s(a, c)$. 又因 $A/B/C$ 为相邻晶圆,共享设备越多,资源冲突越大,则在共享设备累积驻留时间越长,即

$$\sum_{i=1}^{\varphi(a, b)} T_{\text{delay}}(PM_{ab}^i) \leq \sum_{i=1}^{\varphi(a, c)} T_{\text{delay}}(PM_{ac}^i).$$

式中 PM_{ab}^i 为晶圆 J_a 与晶圆 J_b 第 i 个共享加工模块, $T_{\text{delay}}(PM_{ab}^i)$ 为晶圆 J_a (晶圆 J_b)由于等待晶圆 J_b (晶圆 J_a) 在 PM_{ab}^i 加工而发生的驻留时间。

定义 2 蚁群期望启发因子 η_{ij} 表示路径 (i, j) 的能见度,反映晶圆 J_i 转移到晶圆 J_j 的启发程度,这个量在系统运行中不改变,为了尽量避免相邻晶圆流的冲突,根据定理 1,令 $\eta_{ij} = DFP_{ij}$.

定义 3 蚁群信息启发因子 $\tau_{ij}(t)$ 表示 t 时刻路径 (i, j) 上的信息素轨迹强度。

定义 4 蚂蚁搜索得到的一条运动路径即为一个晶圆初始进入顺序方案,用禁忌表 $\text{tabu}_k(s)$ 记录 k 为蚂蚁编号, s 为晶圆加工序号。

本文 AS&TC 算法的核心思想包括两个阶段:外层晶圆顺序决策和内层机械手调度。

算法的具体步骤描述如下:

步骤 1 产生初始晶圆进入顺序。

步骤 1.1 将 m 只蚂蚁随机的置于 n 片晶圆节点上,赋值 $s = 1$, $\text{tabu}_k(s) = \text{random}(1, n)$, $k = 1, 2, \dots, m$, 对 n 片晶圆进行编号,为 J_1, J_2, \dots, J_n ,

作为蚂蚁路径搜索初始节点。

步骤 1.2 按照蚁群算法状态转移概率式(18)和式(19)选取下一片晶圆 j ,并更新禁忌表。

赋值 $s = s + 1$,产生随机数 $q = \text{random}(0, 1)$,

$$j = \begin{cases} \arg \max_{u \in \text{allowed}_k} \{\tau_{ij}^\alpha(t) \eta_{ij}^\beta\}, & q \leq q_0; \\ J, & q > q_0. \end{cases} \quad (18)$$

q_0 为转移判定系数, $q_0 \sim U(0, 1)$, J 由式(19)确定,当 $q > q_0$ 则 j 由转移概率 $P_{ij}^k(t)$ 确定。

$$P_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta}{\sum_{s \in \text{allowed}_k} \tau_{is}^\alpha(t) \eta_{is}^\beta}, & j \in \text{allowed}_k; \\ 0, & \text{其他}. \end{cases} \quad (19)$$

步骤 1.3 信息素局部更新。

$$\tau_{ij}(t) = (1 - \rho) \tau_{ij}(t) + \rho \cdot \tau_0. \quad (20)$$

其中 $1 - \rho$ 为信息素挥发系数, τ_0 为信息素初始值。

步骤 1.4 重复步骤 1.2, 1.3 直至产生完整的晶圆进入顺序。

步骤 2 正向查找可用时间区间。

步骤 2.1 根据 x_i 的位置类型,计算晶圆在相应加工模块可行开始区间。

1) 假设 x_{i-1} 位于 C^{h-1} ($i > 1$) 则使用式(21)计算 $\text{ts}_{x_i d}(w)$:

$$\text{ts}_{x_i d}(w) = (((((t_{\text{pm } x_i}(w) \oplus [0, -P_{x_i}(w) - t_{\text{UD}} - t_{\text{LD}}]) \cap (T_{\text{SB } h-1} \oplus [t_M, t_M])) \oplus [-t_M, -t_M]) \cap (T_{R h} \oplus [0, -t_M])) \oplus [t_M, t_M]). \quad (21)$$

式中 $\text{ts}_{x_i d}(w)$ 是处理模块 x_i 的第 d 个可行的开始时间区间, $t_{\text{PM } hj d}(w)$ 是 PM_{hj} 的第 d 个可行的时间区间, $t_{\text{LD}}, t_{\text{UD}}$ 分别为晶圆装载和卸载时间。

3) 假设 x_{i-1} 位于 C^{h+1} ($i > 1$) 则使用式(22)计算 $\text{ts}_{x_i d}(w)$:

$$\text{ts}_{x_i d}(w) = (((((t_{\text{PM } x_i}(w) \oplus [0, -P_{x_i}(w) - t_{\text{UD}} - t_{\text{LD}}]) \cap (T_{\text{SB } h+2} \oplus [t_M, t_M])) \oplus [-t_M, -t_M]) \cap (T_{R h} \oplus [0, -t_M])) \oplus [t_M, t_M]). \quad (22)$$

4) 其他情况,则转至步骤 2.4。

步骤 2.2 使用式(23)计算可行结束时间区间 $\text{tl}_{x_i d}(w)$:

$$\text{tl}_{x_i d}(w) = ((t_{\text{PM } hj d}(w) \oplus [P_{hj}(w), 0]) \cap (\text{ts}_{x_i d}(w) \cup [P_{hj}(w), P_{hj}(w) + U_{hj}(w)]). \quad (23)$$

步骤 2.3 若 $\text{ts}_{x_i d}(w) = \phi$ or $\text{tl}_{x_i d}(w) = \phi$, 则令 $d = d + 1$,并返回步骤 2.1,否则转到下一步。

步骤 2.4 使用式(24)计算晶圆在相应加工模块可行开始时间约束集 $\text{TS}_{x_i}(w)$:

$$\text{TS}_{x_i}(w) = (((T_{\text{PM } x_i}(w) \oplus [0, -P_{x_i}(w) - t_{\text{UD}} - t_{\text{LD}}]) \cap (T_{L_{x_{i-1}}}(w) \oplus [T_M, T_M])) \cap [-T_M, -$$

$$T_M] \cap (T_{R_{jk}}(w) \oplus [0, -t_M]) \oplus T_M] T_M]. \quad (24)$$

步骤 2.5 使用式(25)计算晶圆在相应加工模块可行结束时间约束集 $TL_{x_i}(w)$:

$$TL_{x_i}(w) = tl_{x_{i-1}}(w) \cup tl_{x_{i-2}}(w) \cup \dots \quad (25)$$

步骤 2.6 若 $TS_{x_i}(w) = \phi$ or $TL_{x_i}(w) = \phi$, 则令 $d = d + 1$, 并返回步骤 2.1, 否则转到下一步.

步骤 3 后向求解调度时间点.

步骤 3.1 使用式(26)计算晶圆的最早的加工结束时间 $S_{CMB}(w)$:

$$S_{CMB}(w) = \min T_{S_{CMB}}(w). \quad (26)$$

步骤 3.2 使用式(27)计算 $L_{x_{q(w)}}(w)$:

$$L_{x_{q(w)}}(w) = S_{CMB}(w) - t_M. \quad (27)$$

步骤 3.3 使用式(28)计算 $S_{x_{q(w)}}(w)$:

$$S_{x_{q(w)}}(w) = \max [L_{x_{q(w)}}(w) - P_{x_{q(w)}}(w) - U_{x_{q(w)}}(w), L_{x_{q(w)}}(w) - P_{x_{q(w)}}(w)] \cap T_{S_{x_{q(w)}}}(w). \quad (28)$$

步骤 3.4 使用式(29)计算 $LB_{hj}(w)$:

$$LB_{hj}(w) = S_{x_{i+1}}(w) - t_M. \quad (29)$$

步骤 3.5 使用式(30)计算 $SB_{hj}(w)$:

$$SB_{hj}(w) = \min [0, SB_{hj}(w) - t_M] \cap T_{SB_{hj}}(w). \quad (30)$$

步骤 3.6 使用式(31)计算 $L_{x_i}(w)$:

$$L_{x_i}(w) = S_{x_{i+1}}(w) - t_M. \quad (31)$$

步骤 3.7 使用式(32)计算 $S_{x_i}(w)$:

$$S_{x_i}(w) = \max (([L_{x_i}(w) - P_{x_i}(w) - U_{x_i}(w), L_{x_i}(w) - P_{x_i}(w)]) \cap T_{S_{x_i}}(w)). \quad (32)$$

步骤 3.8 令 $i = i - 1$; 若 $i = 0$, 后向搜索结束, 否则转至步骤 3.4.

步骤 4 数据信息更新.

步骤 4.1 更新搬运模块可用时间集数据:

$$T_{R_{jk}}(w+1) = \delta_{T_{R_{jk}}(w)}([t_{RS_{x_1}}(w) \setminus L_{S_{x_1}}(w)] \cup \dots \cup [t_{RBS_{hj}}(w) \setminus LBS_{hj}(w)] \cup \dots \cup [t_{RS_{x_q}}(w) \setminus L_{S_{x_q}}(w)]). \quad (33)$$

步骤 4.2 更新各加工模块的可用时间集数据:

$$T_{PM_{hj}}(w+1) = \delta_{T_{PM_{hj}}(w)}([S_{hj}(w) \setminus L_{hj}(w)]). \quad (34)$$

步骤 4.3 各缓冲模块可用时间集数据:

$$T_{BM_{hj}}(w+1) = \delta_{T_{BM_{hj}}(w)}([SB_{hj}(w) \setminus LB_{hj}(w)]). \quad (35)$$

步骤 5 令 $s = s + 1$; 若 $s \leq n$, 则返回步骤 2, 否则转至下一步.

步骤 6 初始化集束型设备群, 令 $k = k + 1$; 若 $k \leq m$, 则返回步骤 2, 否则转至下一步.

步骤 7 信息素全局更新.

$$\tau_{ij}(t) = (1 - \alpha_1) \tau_{ij}(t) + \alpha_1 \cdot \Delta \tau_{ij}(t). \quad (36)$$

其中 $\Delta \tau_{ij}(t) = \begin{cases} (L_{\min})^{-1} & (i, j) \in \text{全局最优路径}; \\ 0, & \text{其他}. \end{cases}$

式中: $1 - \alpha_1$ 表示信息素全局挥发系数; $L_{\min} =$

$\min\{L_1, L_2, \dots, L_m\}$; L_1, L_2, \dots, L_m 表示蚂蚁 $1 \sim m$ 搜索到的晶圆序列调度得到的 Makespan.

步骤 8 令循环次数 $N_c = N_c + 1$; 若 $N_c \leq N_{\max}$, 则返回步骤 1, 否则算法结束.

3 仿真实验及分析

现将实验相关的符号与变量定义如下.

$R = (C_{ASTC} - C_{SPT}^*) / C_{SPT}^* \times 100\%$ 为制造期相对延时率, 表示算法相对于 C_{SPT}^* 的延迟, 该值越小, 表明调度得到的制造期越接近 C_{SPT}^* , 即算法的性能越好. C_{ASTC} 表示本文算法处理一批晶圆的 Makespan. C_{SPT}^* 表示文献[11]中提出的初始晶圆进入顺序按 SPT(shortest processing time)排序的情况下处理一批晶圆的 Makespan. $df = k/n$, 为晶圆流批异化率, 即批量为 n 的批中具有 k 种不同的晶圆种类的比例. 实验用 C++ 实现, 仿真环境为主频 2.13 GHz, 内存 8 GB 的 PC 机. 算法相关基本参数设定: $m = 25$, $\alpha = 2$, $\beta = 5$, $\alpha_1 = 0.1$, $\rho = 0.1$, $q_0 = 0.9$.

3.1 运算时间分析

晶圆处理时间服从 $P \sim N(40, 10)$, 驻留约束时间服从 $U \sim N(20, 5)$. 集束型设备数目 $CT = 3$; 每个集束型设备的加工模块数目 $MH = 4$, $t_M = 4$; $df = 0.6$. 仿真运行的结果如图 2 所示.

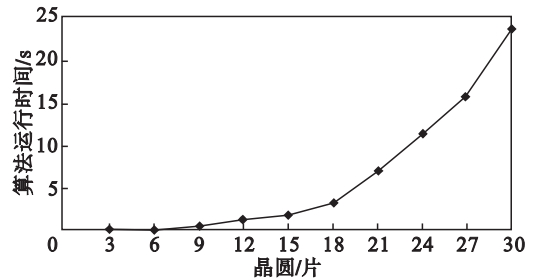


图 2 算法运行时间分析

Fig. 2 The running time of algorithm

图 2 显示了算法的运行时间随着晶圆数目的增加大体上呈正相关, 在晶圆数目较少时(小于 10 片)显然算法的运算时间是非常小的, 当晶圆数目 $N = 9$ 时, 运算时间仅为 0.39 s. 当晶圆数目增加到 20 片以上时, 算法运算时间较长.

3.2 晶圆流对调度的影响分析

令 $CT = 3$, $MH = 4$, $t_M = 4$, $df = 0.4, 0.6, 0.8, 1$. 处理时间服从 $P \sim N(40, 10)$, $U \sim N(20, 5)$. 仿真运行的结果如图 3 所示.

图 3 显示不同晶圆流批异化率情况下, 随着加工晶圆数量增加, R 值呈现先上升后逐渐趋于稳定. 实际工程应用中, 一般批量设置为 25 片左

右,右半段(批量大于 10)更具参考价值。

3.3 处理时间对调度的影响分析

CT=3 ;MH=4 ; $t_M=4$;df=0.6 ;处理时间分别服从 $P \sim N(20,10)$, $P \sim N(40,10)$, $P \sim N(60,10)$, $U \sim N(30,5)$ 。结果见图 4。

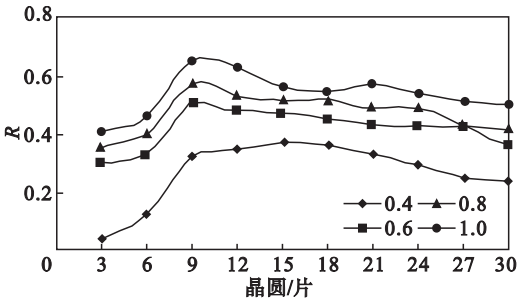


图 3 不同晶圆流批异化率下 R 与晶圆数目的关系
Fig. 3 The relationship between R and N with different df

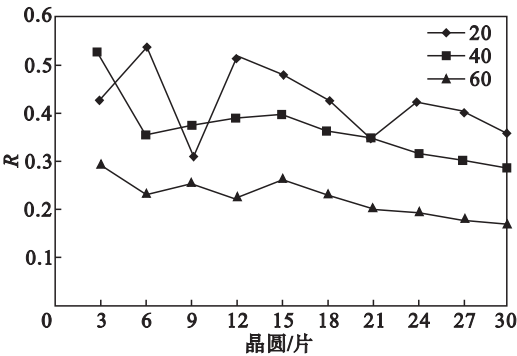


图 4 处理时间波动下 R 与晶圆数目的关系
Fig. 4 The relationship between R and N with different P

图 4 显示在处理时间波动的情况下 R 与晶圆数目的关系。由于处理时间与搬运时间差别较小, $E(P)=20$ 曲线波动较大,图 4 中 $E(P)=40$ 和 $E(P)=60$ 的两条曲线的走向趋势基本一致的,说明算法对不同驻留约束时间的适应性。

3.4 驻留时间对调度的影响分析

令 CT=3 ;MH=4 ; $t_M=4$;df=0.6 ; $P \sim N(40,10)$ 。驻留约束时间分别服从 $U \sim N(10,5)$, $U \sim N(20,5)$, $U \sim N(30,5)$ 。结果见图 5。

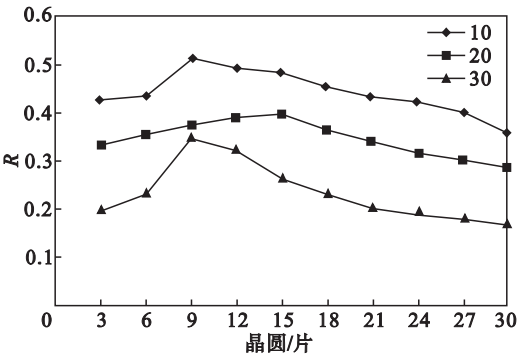


图 5 驻留约束时间波动下 R 与晶圆数目的关系
Fig. 5 The relationship between R and N with different U

图 5 显示在驻留约束时间波动的情况下 R 与晶圆数目的关系。一般情况下,驻留约束时间越短,调度越严格,所需加工时间可能越长,图 5 的结果与此相符。随着每批晶圆数目增大, R 不断减小,说明该算法在批量较大时有较好的调度效果。

4 结 论

- 1) 本文算法可以有效地解决单臂 MCTs 调度中存在的多品种加工以及晶圆驻留问题,特别为客户需求多样化问题提供了良好的解决方案。
- 2) 批量较小(每批少于 10 片)得出算法仿真运行时间非常小,可有效地进行实时调度,即使批量较大也满足调度的要求,证明了算法的高效性。
- 3) 对于晶圆加工时间、驻留时间波动性和晶圆流模式的仿真证明算法对不同的晶圆加工数据具有非常好的适应性。

参考文献：

[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] Yoon H J ,Lee D Y. On-line scheduling of integrated single wafer processing tools with temporal constraints[J] . *IEEE Transactions on Semiconductor Manufacturing* 2005 ,18(3) :390 - 398.

[3] Zhou B H ,Li X. Try and error-based scheduling algorithm for cluster tools of wafer fabrications with residency time constraints[J] . *Journal of Central South University of Technology* 2012 ,19(1) :187 - 192.

[4] Zhou B H ,Gao Z S ,Chen J. Scheduling algorithm for dual-arm cluster tools with residency and reentrant constraints[J] . *Journal of Central South University of Technology* 2014 ,21(1) :160 - 166.

[5] Yi J ,Ding S ,Song D ,et al. Steady-state throughput and scheduling analysis of multi-cluster tools :a decomposition approach[J] . *IEEE Transactions on Automation Science and Engineering* 2008 ,5(2) :321 - 336.

[6] Chan W K ,Yi J ,Ding S. Optimal scheduling of multi-cluster tools with constant robot moving times ,part I :two-cluster analysis[J] . *IEEE Transactions on Automation Science and Engineering* 2011 ,8(1) :5 - 16.

[7] Chan W K ,Yi J ,Ding S. Optimal scheduling of multicluster tools with constant robot moving times ,part II :tree-like topology configurations [J] . *IEEE Transactions on Automation Science Engineering* 2011 ,8(1) :17 - 28.

[8] 石萧铭 ,周炳海. 带驻留约束及双臂机械手的集束型设备群调度算法[J] . *上海交通大学学报* ,2013 ,47(4) :650 - 655.

(Shi Xiao-ming ,Zhou Bing-hai. Scheduling algorithm for dual-arm multi-cluster tools with residency constraints[J] . *Journal of Shanghai Jiaotong University* ,2013 ,47(4) :650 - 655.)

[9] Liu M X ,Zhou B H. Modeling and scheduling analysis of multi-cluster tools with residency constraints based on time constraint sets[J] . *International Journal of Production Research* 2013 ,51(16) :4835 - 4852.

[10] Dechter R ,Meiri I ,Pear J. Temporal constraint networks[J] . *Artificial Intelligence* ,1991 ,49(1/2/3) :61 - 95.

[11] Kim H J ,Lee J H ,Kim C H ,et al. Scheduling in-line multiple cluster tools a decomposition approach[C] // *IEEE International Conference on Mechatronics and Automation*. Chengdu 2012 :1544 - 1549.