

考虑多约束的 MOJ 调度问题

周炳海, 王 腾, 方 腾

(同济大学 机械与能源工程学院, 上海 201804)

摘 要: 统筹考虑晶圆加工过程中的多品种、p-s-d(past-sequence-dependent)换模时间及衰退效应等约束特征,以总加权提前/拖期惩罚成本最小为优化目标,建立了单机 MOJ(multiple orders per job)调度数学规划模型.在此基础上,对决策变量进行分离,提出具有双层嵌套编码机制的改进型遗传蚁群调度算法.该算法将遗传算法融合到动态自适应蚁群算法的每一次迭代过程中,并为有效提高算法的收敛性能,引入 ATCS(apparent tardiness cost with setups)修正准则.最后,仿真实验结果表明,该算法是有效、可行的.

关 键 词: 多品种;p-s-d 换模时间;衰退效应;调度;改进型遗传蚁群算法

中图分类号: TP 391 **文献标志码:** A **文章编号:** 1005-3026(2015)10-1506-06

Scheduling Multiple Orders per Job with Various Constraints

ZHOU Bing-hai, WANG Teng, FANG Teng

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

Abstract: Taking a comprehensive consideration of the characteristics of multiple product types, the past-sequence-dependent (p-s-d) setup time and the deterioration effects constraints in processes of wafer fabrication, with an objective function of minimizing total weighted earliness-tardiness penalties cost, a mathematical programming model of scheduling multiple orders per job (MOJ) in a single machine was built. On this basis, the decision-making variables were separated, and a modified genetic algorithm-ant colony optimization (MGA-ACO) algorithm adopting two-level encoding mechanism was put forward. Genetic algorithm was converged to the process of dynamic and adaptive ant colony iterations. To improve the algorithm convergence performance, a modified rule of apparent tardiness cost with setups (ATCS) was applied. Finally, the simulation results indicated that the developed algorithm is valid and feasible.

Key words: multiple product types; p-s-d setup time; deterioration effects; scheduling; MGA-ACO algorithm

随着 300 mm 晶圆加工技术的普及,单作业多订单(multiple orders per job, MOJ)调度问题^[1]作为半导体晶圆生产线上常见的 NP-hard 问题^[2],近年来受到了业界和学术界越来越多的关注.

为有效解决单机 MOJ 调度问题,学者们已提出了多种调度方法. Qu 等^[1]对单一品种、不考虑到达时间等因素的理想化单机 MOJ 调度模型进行了基于禁忌搜索算法的调度方法研究. Sobeyko 等^[2]对理想化单机 MOJ 调度模型,构造了分组遗传算法和随机键遗传算法. Jampani 等^[3]对带准

备时间的单机 MOJ 调度问题,构造了列生成启发式算法. 上述文献提及的调度方法,探讨的问题规模均不超过 100 个客户订单,而 Mason 等^[4]提出的启发式算法对订单规模数为 100 以上甚至达到 10 000 的超大规模单机 MOJ 调度问题进行了有效求解.

在单机 MOJ 调度研究领域,现有 Erramilli 等^[5-6]在批调度研究中考虑了产品的多样性. 但批调度的研究重点是对由不同作业形成的批次的调度,与本文探讨的调度问题有较大区别.

Chen^[7]对考虑多品种、与次序相关的换模时

收稿日期: 2014-03-12

基金项目: 国家自然科学基金资助项目(71471135, 61273035).

作者简介: 周炳海(1965-),男,浙江浦江人,同济大学教授,博士生导师.

间的流水线车间 MOJ 调度问题,建立了混合 0-1 整数规划模型,但未考虑设备的衰退效应. 对带衰退效应的 MOJ 调度问题进行研究,具有普遍性和实用性,而目前鲜有文献探讨衰退效应背景下的 MOJ 调度方法.

本文研究的单机 MOJ 调度问题以最小化总加权提前/拖期惩罚成本为目标函数,同时考虑订单品种的多样性, p-s-d (past-sequence-dependent) 换模时间以及设备的衰退效应.

1 问题描述

单机 MOJ 调度问题包含订单成组形成作业和作业排序两个子问题,现举例说明,见图 1. 图 1 中有包含两种晶圆①和②的订单 O_1, O_2, \dots, O_5 和承载上限值为 3 片晶圆的作业 J_1, J_2, J_3 . 首先进行订单分组形成作业 $J_1 = \{O_1\}, J_2 = \{O_2, O_3, O_4\}, J_3 = \{O_5\}$; 然后对分配好订单的作业 J_1, J_2, J_3 进行排序,图 1 中假设进入机器加工的作业顺序为 J_2, J_3, J_1 .

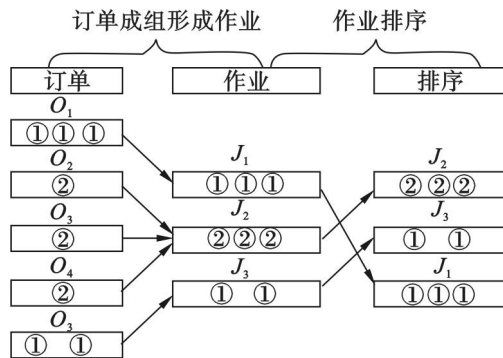


图 1 单机 MOJ 调度问题实例

Fig. 1 Instance for single machine MOJ scheduling problem

为有效表述单机 MOJ 调度问题,在文献[7]的基础上,现做如下假设:①调度过程中,订单需保持完整;②每个订单只含有一种产品,不同订单的产品种类可以不同;③每个作业只含有产品种类相同的订单;④每个作业中载有的晶圆数不能超过其承载上限 K ;⑤作业的数量能够满足调度要求且参与调度的作业能被有效利用;⑥作业加工时间随时间的恶化过程为非定态伽马过程;⑦机器在加工不同品种产品时考虑 p-s-d 换模时间;⑧机器一旦开始工作,除换模外,不会中断;⑨同一个作业中的订单有相同的完工时间.

对相关符号定义如下,以更清晰地描述调度问题: M 为一个较大的正数; N 为订单数; K 为晶圆的承载上限值; φ 为产品种类数; ϕ 为作业(基

因或节点)个数; λ_{il} 为如果订单 i 的类型属于产品 l , 则其值为 1, 反之为 0; f_i 为订单 i 的产品种类标号; n_i 为订单 i 中含有晶圆的数量; d_i 为订单 i 的交货期; w_i 为订单 i 的权重; ρ^f 为批加工方式下订单 i 单位晶圆所需的加工时间; a 为单位提前时间的惩罚金额; b 为单位延迟时间的惩罚金额 ($b > a$); γ 为 p-s-d 换模时间的规模系数; β 为非定态伽马过程的尺度参数; α 为非定态伽马过程的形状参数; q 为非定态伽马过程的幂指数; p_j 为作业 j 的实际加工时间; σ 为在作业 j 之前加工的作业的序号; h_j 为加工作业 j 时所需的换模时间; e_j 为作业 j 的恶化加工时间; C_i 为订单 i 的完工时间; s_j 为作业 j 的开始加工时间; X_{ij} 为如果订单 i 被分配到了作业 j , 则其值为 1, 反之为 0; U_j 为如果有订单分配给作业 j , 则其值为 1, 反之为 0; $v_{jj'}$ 为如果作业 j 和作业 j' 产品种类相同, 则其值为 0, 否则为 1; E_i 为订单 i 的提前完工时间; T_i 为订单 i 的拖期完工时间. 由假设①可知, 一个客户订单对应只能分配到一个作业中, 即需满足式(1):

$$\sum_{j=1}^{\phi} X_{ij} = 1, i = 1, 2, \dots, N. \quad (1)$$

由假设②和③可知, 只有产品相同的订单才能分配到同一个作业中, 即需同时满足式(2)~式(4):

$$f_i = \sum_{l=1}^{\varphi} l \lambda_{il}, i = 1, 2, \dots, N; \quad (2)$$

$$f_i \leq f_{i'} + M(2 - X_{ij} - X_{i'j}), i, i' = 1, 2, \dots, N, \text{ 且 } i \neq i'; j = 1, 2, \dots, \phi; \quad (3)$$

$$f_i \geq f_{i'} - M(2 - X_{ij} - X_{i'j}), i, i' = 1, 2, \dots, N, \text{ 且 } i \neq i'; j = 1, 2, \dots, \phi. \quad (4)$$

由假设④易知, 每个作业中所有订单的晶圆总数不能超过 K , 即需满足式(5):

$$\sum_{i=1}^N n_i X_{ij} \leq K, j = 1, 2, \dots, \phi. \quad (5)$$

由假设⑤可知, 参与调度的每一个作业, 其含有的晶圆个数不能为 0, 即需同时满足式(6), 式(8):

$$\sum_{i=1}^N X_{ij} < MU_j, j = 1, 2, \dots, \phi; \quad (6)$$

$$U_j < M \sum_{i=1}^N X_{ij}, j = 1, 2, \dots, \phi; \quad (7)$$

$$U_{j+1} \leq U_j, j = 1, 2, \dots, \phi - 1. \quad (8)$$

在假设⑥和文献[8]的基础上, 得到作业恶化加工时间的计算需满足式(9)~式(11):

$$e_j = \alpha \beta s_j^q, j = 1, 2, \dots, \phi; \quad (9)$$

$$s_1 = 0; \quad (10)$$

$$s_j = \sum_{\sigma=1}^{j-1} p_{\sigma}, j = 2, \dots, \phi. \quad (11)$$

在假设⑦和文献[9]的基础上可知,换模时间需满足式(12),式(13):

$$h_1 = 0, \quad (12)$$

$$h_j = \gamma v_{jj'} s_j, j = 2, \dots, \phi, j' = j - 1. \quad (13)$$

在假设⑧的基础上可以得到批加工方式下每个作业的实际加工时间,计算公式为

$$p_j = \frac{\sum_{i=1}^N X_{ij} \rho^{f_i}}{\sum_{i=1}^N X_{ij}} + h_j + e_j, j = 1, 2, \dots, \phi. \quad (14)$$

由假设⑨知,订单完工时间的计算需满足式(15):

$$C_i = \sum_{j=1}^{\phi} (s_j + p_j) X_{ij}, i = 1, 2, \dots, N. \quad (15)$$

订单的提前期或者拖延期由其完工时间决定,计算公式分别为

$$E_i = \max(0, d_i - C_i), i = 1, 2, \dots, N; \quad (16)$$

$$T_i = \max(0, C_i - d_i), i = 1, 2, \dots, N. \quad (17)$$

调度目标为总加权提前/拖期惩罚成本最小,即

$$\min \sum_{i=1}^N (aw_i E_i + bw_i T_i). \quad (18)$$

因此,研究的单机 MOJ 调度问题是以式(18)为目标,以式(1)~式(17)为约束的非线性规划问题.

2 算法构建

将遗传算法融合到动态自适应调整信息素的蚁群算法的迭代过程中,为改善算法的寻优性能,构造 ATCS (apparent tardiness cost with setups) 准则进行检验和修正,从而建立改进型遗传蚁群算法(modified genetic algorithm - ant colony optimization algorithm, MGA - ACO). 为方便描述,现做如下定义.

定义 1 $O_j = (X_{1j}, 2X_{2j}, \dots, NX_{Nj}), j \in \{1, 2, \dots, \phi\}$ 表示由分配到作业 j 的订单序号组成的一维数组.

定义 2 蚂蚁的搜索节点或者遗传操作中的基因 $\pi_j, j \in \{1, 2, \dots, \phi\}$ 指调度过程中形成的可行作业.

定义 3 蚂蚁搜索得到的一条运动路径或者遗传操作中的一条染色体即为一个可行调度方案,用禁忌表 $\text{Tabu}_m, m \in \{1, 2, \dots, \phi\}$ 记录.

定义 4 $g(\text{Tabu}_m), m \in \{1, 2, \dots, \phi\}$ 为运动

路径(染色体) Tabu_m 的目标函数值,其中

$$g(\text{Tabu}_m) = \sum_{i=1}^N (aw_i E_i + bw_i T_i). \quad (19)$$

算法的核心操作主要为编码、初始化算子、遗传操作和修正算子. 具体步骤如下.

步骤 1 编码:分离数学模型中的决策变量,采用双层嵌套编码机制,下层的节点(基因)编码由约束条件得到初始解,并由遗传算法的变异算子搜索,上层的作业加工序号主要由蚁群算法和 ATCS 修正准则编码. 上、下两层编码通过作业序号形成映射关系. 为满足订单、作业序号的标记需求,简便算法操作,采用自然数编码.

步骤 1.1 节点(基因)编码:依据订单分配原则(见式(1)~式(8))可得蚂蚁搜索节点或遗传基因 $\pi_j = (j, f_j, O_j)$. 节点(基因)集合为 $\Gamma = \{\pi_j | j = 1, 2, \dots, \phi\}$. 图 2 为对图 1 实例的编码示例.

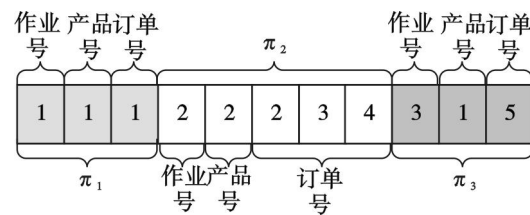


图 2 节点(基因)编码示例

Fig. 2 Example for coding node (gene)

步骤 1.2 路径(或染色体)编码:将 ϕ 只蚂蚁随机放置在 ϕ 个搜索节点上,蚂蚁 $m, m \in \{1, 2, \dots, \phi\}$ 在时刻 t 从节点 j 转移到节点 $j' (j' \neq j)$ 的概率为

$$p_{jj'}^m(t) = \begin{cases} \frac{[\tau_{jj'}(t)]^{\alpha_1} [\eta_{jj'}(t)]^{\beta_1}}{\sum_{u \in \text{allowed}_m} [\tau_{ju}(t)]^{\alpha_1} [\eta_{ju}(t)]^{\beta_1}}, & \text{若 } j' \in \text{allowed}_m; \\ 0, & \text{否则.} \end{cases} \quad (20)$$

式中: $\text{allowed}_m = \{\Gamma - \text{Tabu}_m\}$ 表示蚂蚁 m 下一步允许选择的节点; $\tau_{jj'}(t)$ 是时刻 t 路径 (j, j') 上的信息量; α_1, β_1 分别为信息启发因子和期望启发因子; $\eta_{jj'}(t)$ 为期望启发函数:

$$\eta_{jj'}(t) = \left(\frac{\sum_{i=1}^N X_{ij'} (aw_i E_i + bw_i T_i)}{\sum_{u \in \text{allowed}_m} \sum_{i=1}^N X_{iu} (aw_i E_i + bw_i T_i)} \right)^{-1}. \quad (21)$$

在计算式(9)~式(17),式(20),式(21)的基础上,蚂蚁 m 遍历完所有节点后得到运动路径

(染色体) $\text{Tabu}_m = \{\text{Tabu}_m[j] \mid j = 1, 2, \dots, \phi\}$, 其中 $\text{Tabu}_m[j]$ 表示第 j 个被记录下的节点 (基因) 序号. 图 3 为与图 1、图 2 对应的一个路径 (染色体) 编码示例 Tabu_m .

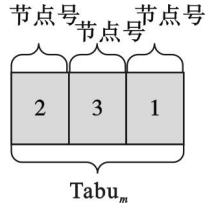


图 3 路径 (染色体) 编码示例

Fig. 3 Example for coding route (chromosome)

通过图 2 和图 3 的映射关系, 得到蚂蚁 m 搜索得到的可行调度方案为 $\{\pi_2, \pi_3, \pi_1\}$.

双层嵌套编码机制, 相比近年由 Sobeyko 等^[2]和 Qu 等^[10]提出的将作业与订单信息合并并在同一层的繁冗编码方式更清晰、简便.

步骤 2 外部循环终止判断: 如果蚂蚁搜索的循环次数 θ_1 达到上限值 θ , 则输出结果, 结束运算, 否则, 进入步骤 3.

步骤 3 初始化算子: ϕ 只蚂蚁通过步骤 1. 2 遍历完 ϕ 个节点后, 得到初始种群 $\text{population} = \{\text{Tabu}_m \mid m = 1, 2, \dots, \phi\}$.

步骤 4 适应度评估: 基于式 (18) 和式 (19), 得到适应度函数:

$$\begin{aligned} \text{fit}(\text{Tabu}_m) &= \max(g(\text{Tabu}_1), g(\text{Tabu}_2), \dots, \\ &g(\text{Tabu}_m), \dots, g(\text{Tabu}_\phi)) - g(\text{Tabu}_m), m \in \{1, 2, \dots, \phi\}. \end{aligned} \quad (22)$$

由式 (9) ~ 式 (17)、式 (19) 和式 (22) 计算出初始种群中每一个个体的适应度.

步骤 5 内部迭代终止判断: 如果遗传操作的迭代次数 ε_1 达到规定上限 ε , 则各路径进行如下信息素更新:

$$\begin{aligned} \tau_{jj'} &= (1 - \psi)\tau_{jj'} + \sum_{m=1}^{\phi} \Delta\tau_{jj'}^m, j, j' \in \{1, 2, \dots, \phi\} \text{ 且 } j' \neq j; \\ \Delta\tau_{jj'}^m &= \begin{cases} \frac{Q_{\theta_1}}{g(\text{Tabu}_m)}, & \text{若蚂蚁 } m \text{ 经过路径 } (j, j'); \\ 0, & \text{否则.} \end{cases} \end{aligned} \quad (23)$$

式中: $\psi \in (0, 1)$ 为信息挥发系数; $\Delta\tau_{jj'}^m$ 为本次循环 θ_1 中蚂蚁 m 留在路径 (j, j') 上的信息量; Q_{θ_1} 为至 θ_1 次循环为止得到的最优目标函数值, 以动态自适应调整信息素. $\theta_1 = \theta_1 + 1$, $\varepsilon_1 = 1$, 回到步骤 2, 否则, 进入步骤 6.

步骤 6 遗传操作.

步骤 6.1 选择算子: 依据个体适应度, 采用轮盘赌法选择出父本.

步骤 6.2 多点交叉: 对每一对交叉父本 Tabu_m 和 $\text{Tabu}_{m'} (m' \neq m)$ 采用多点交叉, 生成新的子代.

步骤 6.3 变异算子

步骤 6.3.1 随机选择变异基因 $\pi_j, j \in \{1, 2, \dots, \phi\}$, 此时选择次数 ζ 记为 1.

步骤 6.3.2 搜索品种标号同为 f_j 的其他基因.

步骤 6.3.3 若有符合步骤 6.3.2 的基因 $\pi_{j'}, j' \neq j$, 则随机互换 O_j 和 $O_{j'}$ 中的部分值. 如果得到的新基因满足假设④或者循环次数 ζ 达到上限值 δ , 则进入步骤 7, 否则, 进入步骤 6.3.4.

步骤 6.3.4 随机选择另一个变异基因 $\pi_{j''}, j'' \in \{1, 2, \dots, \phi\}$ 且 $j'' \neq j$, 令 $\zeta = \zeta + 1$, 回到步骤 6.3.3.

步骤 7 修正算子: 在前人研究的 ATCS 准则^[11]的基础上, 针对本文的问题域, 构造了在时刻 t , 未确定加工位置的作业 $j, j \in \{1, 2, \dots, \phi\}$ 所具有的紧急度指数 $I_j(t)$, 批加工方式下的计算公式见式 (25), 设 k_1, k_2 分别为 2 和 1.

$$I_j(t) = \frac{\sum_{i=1}^N X_{ij} \frac{w_i}{\rho^{f_i}} \exp\left(-\frac{\max(d_i - \rho^{f_i} - t, 0)}{k_1 \rho^{f_i}}\right) \exp\left(\frac{1}{k_2}\right)}{\sum_{i=1}^N X_{ij}}. \quad (25)$$

对种群中的个体以一定概率进行如下 ATCS 修正: 按紧急度指数 $(I_{\max}(t_1), I_{\max}(t_2), \dots, I_{\max}(t_\phi))$ 排列对 $\text{Tabu}_m, m \in \{1, 2, \dots, \phi\}$ 进行修正, 其中 $I_{\max}(t_n), n \in \{1, 2, \dots, \phi\}$ 表示 t_n 时刻即 Tabu_m 中的作业 $\text{Tabu}_m[n]$ 开始加工时刻, 所有未确定加工位置的作业中的最大紧急度指数.

计算新种群的适应度, 若子代的适应度更高, 则接纳子代, 否则用父代取代子代. 经过 ATCS 修正形成新种群后, 令 $\varepsilon_1 = \varepsilon_1 + 1$, 再回到步骤 5.

3 仿真实验分析

为有效评价算法的性能, 以求解时间和解的优劣 (performance ratio, PR)^[10] 作为算法评价指标.

$\text{PR} = \frac{V(H, T, Y)}{\text{Best}(Y)}$. 其中: $V(H, T, Y)$ 表示用算法 H 在迭代次数为 T 时对实例 Y 的求解结果; $\text{Best}(Y)$ 表示对实例 Y 求解所能得到的参照最优

解. PR 值越小,表明算法的收敛性能越好.

本文采用 C++ 编程语言实现算法,在主频为 2.2 GHz,内存为 2 GB 的 PC 机上进行仿真实验.

3.1 参数分析

实验参数参照文献[2,8-11]中的参数设计规则设定.在批加工方式下随机生成了 10 个实例,并分别用 MGA-ACO,GA-ACO 和 ACO 三种算法对实例进行求解,得到迭代次数对 PR 值和求解时间(文中的 PR 值和求解时间均为平均值)的影响如图 4 所示.

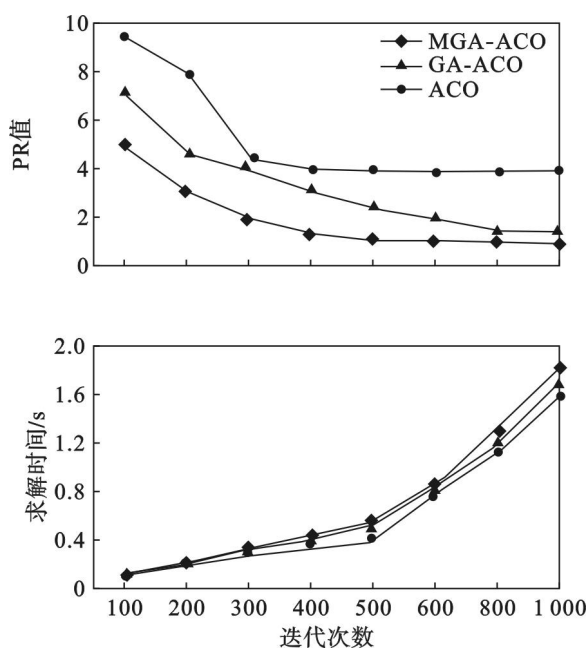


图 4 迭代次数对 PR 值和求解时间的影响

Fig. 4 Effect of iterations on PR and computation time

由图 4 可知,用 MGA-ACO 求解实例所得的 PR 值明显优于 GA-ACO 和 ACO 的结果,而三者的求解时间非常相近,因此由算法评价指标可得,MGA-ACO 的求解性能明显优于 GA-ACO 和 ACO.当迭代次数由 600 增加到 1000 的过程中,MGA-ACO 得到的 PR 值几乎没有改善,但求解时间成本几乎呈线性增加.考虑保证算法的优良求解性能,在后续实验中,将迭代次数选择为 600 次.

3.2 求解时间对调度模型的敏感性分析

算法的求解时间主要由它的时间复杂度决定,算法的时间复杂度越高,其求解时间越长.通过计算得到 MGA-ACO 的时间复杂度为 $O(\varphi^4 + \theta \times \varepsilon \times (N + \varphi))$,由此易知,算法的运算时间与调度模型中的晶圆品种数呈正相关关系,而衰退效应和 p-s-d 换模时间的特征参数值的取值变化不会给算法的运算时间带来影响.

3.3 PR 值对调度模型的敏感性分析

对产品种类数 $\varphi = 5, 6, 7, 8, 9, 10$,而对应的订单数分别为中小规模的 20,30 和 50 及大规模的 100,150 和 200 的调度问题进行仿真实验.在批加工方式下的实验结果见图 5.

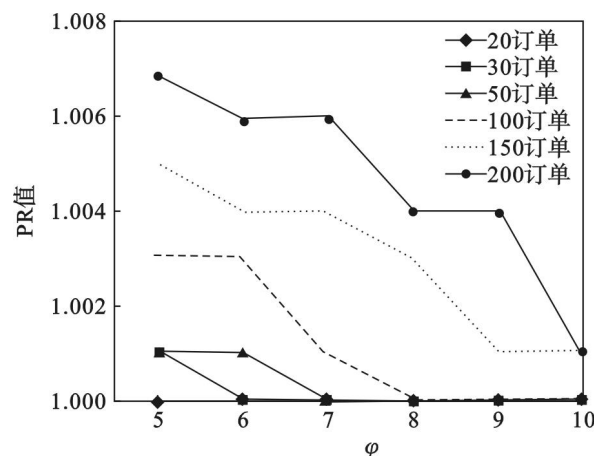


图 5 品种数对 PR 值的影响

Fig. 5 Effect of wafers variety on PR value

由图 5 易知,当订单数相同时,随着品种数的增加,PR 值会随之减小,这与随着品种数增加,订单分配的限制也会增加,调度组合减少,算法更容易求得优秀调度结果相符;当晶圆品种一定时,随着订单数规模的变大,PR 值会随之增大,这与随着订单规模的增大,问题复杂性增强,不易于算法的求解一致.

由实验易得,改变衰退效应和 p-s-d 换模时间的特征参数值后得到的实验结果与图 5 的结果几乎一样.

4 结 论

1) 采用双层嵌套编码机制,以作业序号为纽带及上、下层次间编码信息映射统一的编码方式,为 MOJ 调度问题传统地将所有编码信息冗余集中在同一层次提供了新的研究思路.

2) 将遗传算法融合到动态自适应调整信息素的蚁群算法的迭代过程中,同时构造 ATCS 修正准则,进一步改善算法的收敛性能,建立基于 MGA-ACO 的 MOJ 调度方法,丰富了解决此类调度问题的理论方法.

3) 仿真实验表明,MGA-ACO 对于模型中特征参数的取值变化具有良好的适应性,算法求解结果稳定且令人满意,验证了 MGA-ACO 的可行性和有效性.

(下转第 1520 页)