

基于遗传粒子群混合的可重入生产调度优化

刘小华, 林杰, 邓可

(同济大学 经济与管理学院, 上海 200092)

摘要: 可重入生产调度优化问题是个 NP 难问题, 针对可重入生产调度的特点, 对该优化问题进行数学规划建模, 并通过一些定义将模型映射为有向图, 以便于智能搜索算法的应用. 结合粒子群算法收敛速度快与遗传算法全局搜索能力强的特点, 进行优势互补, 并优化设计相关参数, 构造了一种混合算法. 运用混合算法对供应链优化调度问题模型进行求解, 与标准遗传算法、粒子群算法的求解结果进行比较, 结果表明混合算法有着更好的优化性能.

关键词: 可重入制造; 调度优化; 混合算法; 遗传算法; 粒子群算法

中图分类号: F 273

文献标识码: A

Scheduling Optimization in Re-entrant Lines Based on a GA and PSO Hybrid Algorithm

LIU Xiaohua, LIN Jie, DENG Ke

(College of Economics and Management, Tongji University, Shanghai 200092, China)

Abstract: Scheduling optimization in re-entrant lines proves to be more difficult than in other manufacturing systems, which is well known as a NP-hard problem. A mathematical programming model was established and corresponded with a digraph via several definitions for an intelligent algorithm. A hybrid algorithm was proposed to optimize the objective function, which took the advantages of genetic algorithm and particle swarm algorithm, so this hybrid algorithm integrated global searching ability with high convergence speed. Compared to the results of the normal GA or PSO, simulation results show that the hybrid algorithm is an effective method for scheduling optimization.

Key words: re-entrant lines; scheduling optimization; hybrid algorithm; genetic algorithm; particle swarm algorithm

Kumar 在 1993 年首先提出了可重入生产系统 (re-entrant lines), 并把它作为区别于作业车间和流水车间的第 3 类生产系统^[1], 是指在系统中工件在工艺路线的不同阶段多次访问同一加工站的制造系统. 作为可重入制造系统的典型代表——半导体制造系统, 是半导体加工的关键系统, 具有可重入流、大规模、单件加工设备与批处理加工设备并存、加工周期不确定性以及设备负载不均衡等明显区别于其他制造系统的显著特点, 其控制与调度得到了学术界与工业界的普遍关注^[2].

生产调度优化问题是个 NP 难问题, 可重入生产系统的生产调度由于上述特点以及多订单调度时各子任务之间复杂的工艺约束关系愈加地困难. 事实证明, 随着调度问题规模的扩大, 最优化等传统方法解决此类问题的难度急剧增大, 而智能搜索算法近些年来由于其表现的成效, 成为当前研究热点^[3]. 应用于调度问题的智能搜索方法包括模拟退火、禁忌搜索、遗传算法、蚁群算法和粒子群算法等. 大量的研究表明, 这些算法对于调度问题的解决有着不错的效果, 但是往往有着各自的局限性. 其中粒子群算法使用简单, 收敛速度快, 但是容易早熟, 陷入局部最优^[4]; 而遗传算法则全局搜索能力强, 但搜索速度慢^[5], 两者有较强的互补性.

本文旨在为可重入生产调度优化问题进行数学建模, 并且将遗传和粒子群两种算法进行优势互补, 构造混合算法, 对模型进行求解.

1 问题描述及数学规划模型

1.1 问题描述

以半导体生产线 Mini-fab^[6] 模型为例, 对可重入生产系统调度问题进行说明, 见图 1. 模型共有 3

收稿日期: 2010-01-17

基金项目: 国家自然科学基金重点项目(70531020); 国家“八六三”高技术研究发展计划(2007AA04Z151)

第一作者: 刘小华(1978—), 男, 博士生, 主要研究方向为供应链管理及智能生产系统. E-mail: focus2077@sina.com

通讯作者: 林杰(1967—), 男, 教授, 博士生导师, 管理学博士, 主要研究方向为供应链管理及决策支持系统. E-mail: jielinf@263.com

个设备群,5台设备,分别为 M_1, M_2, M_3, M_4, M_5 ,工件需经过6道工序完成加工.重入出现在所有设备上,第1道工序和第5道工序需在设备 M_1 和 M_2 上加工,第2道工序和第4道工序需在 M_3 和 M_4 上加工,第3道工序和第6道工序需在 M_5 上加工.假设这里有2个工件需要加工,工件1的工序分别表示为 $[O_{11}, O_{21}, O_{31}, O_{41}, O_{51}, O_{61}]$,工件2的工序分别表示为 $[O_{12}, O_{22}, O_{32}, O_{42}, O_{52}, O_{62}]$,这样就存在设备和工序的对应关系: $M_1, M_2 \text{---} O_{11}, O_{12}, O_{51}, O_{52}$; $M_3, M_4 \text{---} O_{21}, O_{22}, O_{41}, O_{42}$; $M_5 \text{---} O_{31}, O_{32}, O_{61}, O_{62}$.

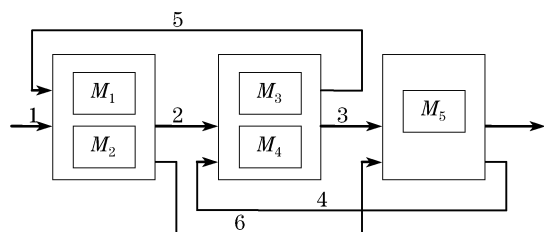


图1 Mini-fab模型

Fig.1 Model of Mini-fab

由上述工序按约束关系组成的序列,就是一个调度方案,可以表示如下: $(O_{11})_{M_1}, (O_{12})_{M_2}, (O_{21})_{M_3}, (O_{22})_{M_4}, (O_{31}, O_{32})_{M_5}, (O_{41})_{M_3}, (O_{42})_{M_4}, (O_{52})_{M_2}, (O_{51})_{M_1}, (O_{61}, O_{62})_{M_5}$.

此序列反映了工件加工的顺序和使用的设备,只要给出起始时间和每道工序的加工时间,就可以得到调度方案的甘特图.但仍然存在以下几个问题:①可以生成的这样的调度方案很多,如何判断解的优劣;②实际生产中,生产线上有上百个不同工艺的产品,每个产品多达300~400个加工工序,由此产生的问题是一个NP难问题;③工序间存在严格的约束关系,特别是由于生产可重入的特性,使这样的约束关系更加复杂.

1.2 数学规划模型

设在时间域值 T_0 内有 M 卡晶圆投入半导体生产线,每卡晶圆按其生产工艺可以分解为 $N_i (i=1, 2, \dots, M)$ 个工序,每卡晶圆的最迟交货期为 $l_i (i=1, 2, \dots, M)$.为便于描述,定义调度方案描述矩阵 C_i 为一个结构矩阵,是描述第 i 卡晶圆各子工序可选方案参数的集合: $C_i = (c_{ijm}) = \langle \langle E, S_{\text{time}}, O_{\text{time}}, C, C_{\text{stock}} \rangle_{ijm} \rangle$.式中, c_{ijm} 表示完成第 i 卡晶圆的第 j 个子工序的第 m 种方案; E 代表执行 c_{ijm} 的加工设备; S_{time} 为 c_{ijm} 的开工时间; O_{time} 为 c_{ijm} 的执行时间; C_{manu} 表示 c_{ijm} 的加工成本; C_{stock} 表示 c_{ijm} 的单位时

间库存成本.以上参数除 S_{time} 外,在进行调度前均为已知.

则建立可重入生产线调度模型如下:

$$\min E = w_1 \left[\sum_{i=1}^M \sum_{j=1}^{N_i} (C_{\text{manu}})_{ij, m_{ij}} + \sum_{i=1}^M \sum_{p=1}^{N_i} \sum_{q=0}^{N_i} [(S_{\text{time}})_{iq, m_{iq}} - (F_{\text{time}})_{ip, m_{ip}}] \cdot k_{ip, q} (C_{\text{stock}})_{ip, m_{ip}} \right] + w_2 \max_i [(F_{\text{time}})_{iN_i, m_{iN_i}}] + P \sum_{i=1}^M \max [(F_{\text{time}})_{iN_i, m_{iN_i}} - l_i, 0] \quad (1)$$

$$\text{s. t. } (S_{\text{time}})_{ij', m_{ij'}} - (S_{\text{time}})_{ij, m_{ij}} \geq (O_{\text{time}})_{ij, m_{ij}}, \quad \text{若 } k_{ij, j'} = 1 \quad (2)$$

式中: $k_{ip, q}$ 为工序约束关系变量,若规则库中工序 p 为 q 的前置工序,则 $k_{ip, q} = 1$,否则 $k_{ip, q} = 0$; w_1, w_2 为相关因素的权重因子; P 为交货期延迟惩罚因子; m_{ij} 表示第 i 卡晶圆的第 j 个子工序最终选定的方案编号; $(F_{\text{time}})_{iN_i, m_{iN_i}}$ 表示第 i 卡晶圆的最后一个子工序的完工时间.

调度模型的第1部分为成本评价项,该部分又包括加工成本和库存成本;第2部分为时间评价项,评估投入的全部卡的完工总费时;第3部分为交货延期惩罚项,以对每卡的交货期进行控制,一旦某卡超过交货期即进行延期惩罚,惩罚因子 P 往往取值较大.将第2,3两部分的时间成本按具体情况折算为费用,以统一量纲.公式(2)为同一卡晶圆各子工序之间的时间约束.

2 遗传粒子群混合算法的设计

2.1 粒子群算法

粒子群优化(particle swarm optimization, PSO)采用群体进化和适应度函数评价优化结果,主要模拟鸟集群飞行觅食行为,通过鸟群的集体协作达到寻优目的.目前为止,研究人员已经开发了多种PSO模型,在本文中使用的是全局优化模型^[7],具体形式如下:

$$V_i = W \times V_i + C_1 \times \text{Rand}() \times (p_{id} - X_i) + C_2 \times \text{rand}() \times (p_{gd} - X_i) \quad (3)$$

$$X_i = X_i + V_i \quad (4)$$

式中: V_i 是微粒 i 的速度,表示该微粒的“飞行”速度; X_i 是微粒 i 的位置; p_{id} 指微粒 i 所经历的最好位置(pbest); p_{gd} 指群体中所有微粒所经历过的最好

位置(gbest); W 为惯性权重,它调整微粒群的全局搜索和局部探索能力; C_1, C_2 为加速度常数,表示把微粒拉向 pbest 和 gbest 的随机项权值; $\text{Rand}()$, $\text{rand}()$ 指 2 个产生于 $[0, 1]$ 范围内的随机数.

2.2 遗传算法

遗传算法(genetic algorithm, GA)是模拟生物在自然环境中的遗传和进化过程而形成的一种自适应全局优化概率搜索算法,它的生物学原理是自然界生物的遗传和进化过程^[8]. 在遗传算法中,一个解向量被称为一个染色体(或称个体),解向量中的分量称为基因. 遗传算法的思想即是:由若干个个体组成一个群体,这个群体不断地进行选择、交叉和变异操作,并且每次都按照“优胜劣汰”的规则将优秀个体的基因遗传到下一代,这样最终在群体中将会得到一个优良的个体,即问题的近似最优解. 基本 GA 算法包括选择算子、交叉算子和变异算子.

2.3 遗传粒子群混合算法

粒子群算法使用简单,收敛速度快,但是容易早熟,陷入局部最优;而遗传算法则全局搜索能力强,但搜索速度慢. 可以看出这两种算法之间有着很强互补性,结合可重入生产调度问题的特点,如果将这两种基于群概念的算法结合对可重入生产调度优化问题进行求解,从而得到一个整体性能更优的混合算法(GA-PSO,简称 GP). 思路就是利用粒子群算法收敛速度快的特点进行前一阶段的优化,得到一定进化程度的初始种群,然后由遗传算法进行后一阶段的优化,避免陷入局部最优. 该算法的优化框架如图 2 所示.

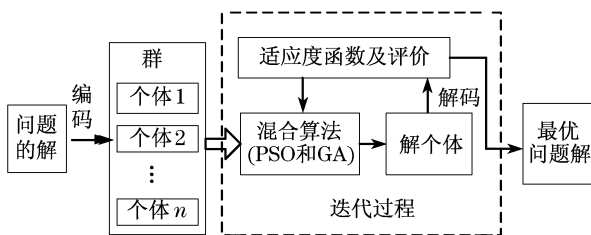


图 2 可重入生产调度优化的流程框架

Fig.2 Process frame of scheduling optimization in re-entrant lines

首先,根据问题的编码规则,随机生成若干个个体编码,组成一个种群. 该种群依据混合算法进行进化,进化过程产生的新个体将被解码,得到对应的目标函数值. 评价机制根据这些目标函数值,对新种群中的个体进行比较和评价,可得到当前的最优解. 与此同时,每一次对种群的评价结果又为算法的进化过程提供方向. 上述过程循环迭代,最终算法得到一

个最优个体.

3 可重入生产调度问题的混合算法求解

3.1 映射及编码

为便于使用智能搜索算法求解,有必要先将模型映射到图中,为此定义有向图 $G = (N, A)$,式中, $N = (n_{ijm})$ 是图中节点的集合,每个节点对应方案描述矩阵 $C_i (i = 1, 2, \dots, M)$ 中的一个元素; A 是图中弧的集合,表示节点间的关系,即规则库中所定义的工序约束关系. 为遵守工序约束关系,对于任意节点 n_{ijm} ,由它出发的弧指向所有满足 $i' \neq i$ 或 $(i' = i$ 且 $k_{ij'j} \neq 1)$ 的 $n_{i'j'm'}$. 以 minfab 模型为例,依然假设只有 2 个工件. 每个工件有 6 道工序,共 12 个子任务. 第 1, 2, 4, 5 道工序可加工设备均有 2 种,即有 2 种加工方案,则共有 20 种可选加工方案,每个可选加工方案对应图 G 中的 1 个节点,共 20 个节点. 由于工序之间的约束关系,所以代表节点关系的弧并非节点间任意连接的,必须满足定义根据工序约束关系确定的条件.

另外,再定义虚拟起始点 sPoint 为图中 1 个仅与子任务约束矩阵 $K_i (i = 1, 2, \dots, M)$ 中全为 0 的列对应子任务节点相连的节点,虚拟起始点不占用任何操作时间及操作成本.

通过上述两定义,使得编码简单便利. 采用实数编码,表示优化问题的解用实数表示更自然,变量的物理意义明确,克服了二进制编码解码的换算占用计算机时间的问题,以及表达精度的要求与计算量之间的矛盾,计算精度不受影响.

以上述 2 个工件为例,每个工件 6 道工序,需要完成的子任务共 12 个,按下式编码:

$$[(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), \dots, (x_{12}, y_{12}, z_{12})]$$

式中: x_i 表示工件序号; y_i 表示工序序号; z_i 表示设备号. 则 (x_i, y_i, z_i) 代表了一个加工方案,整式表示一个调度方案,即为群智能算法中的一个个体.

3.2 算法参数设计

算法的参数对于算法的性能有着较大的影响,因此进行参数的设置是一项重要的内容. 混合算法由粒子群算法和遗传算法两部分结合运用而成,以下分别对这两部分进行参数分析设计,设计结果见表 1.

表1 GA-PSO算法中的参数设计

Tab.1 Parameter design of GA-PSO algorithm

粒子群运算部分		遗传运算部分	
粒子群规模	20	种群规模	20
最大迭代次数	50	最大迭代次数	150
惯性权重 W	1.6	交叉概率	0.7
加速度常数 C_1, C_2	2.0	变异概率	0.05

3.2.1 粒子群算法参数设计

粒子群算法采用式(3)和式(4)表达的全局优化模型,以下结合可重入调度优化问题,对模型中的相关参数进行设计.

(1) 惯性权重 W 、加速度常数 C_1 和 C_2

惯性权重 W 的意义是使微粒保持运动惯性,使其有扩展搜索空间的趋势,有能力探索新的区域^[9]. 本研究中的惯性权重 W 取为 1.5~2.0. 加速度常数 C_1 和 C_2 用于调整 PSO 空间的“弹性”,低加速度值允许微粒在被拉回前在远离目标的区域徘徊,而高加速度值则导致微粒突然冲向或冲过目标区域. 根据以前研究人员的经验,取 C_1 和 C_2 为 2.0.

(2) 最大速度 V_{\max} 和最大位置 X_{\max}

速度 V_i 和位置 X_i 的绝对值有可能很大,使得微粒有可能一下子飞出问题空间^[10]. 因此,应将 V_i 和 X_i 的值限定在一定范围内,即设定最大速度 V_{\max} 和最大位置 X_{\max} . V_{\max} 作为一种约束来控制微粒群的全局搜索能力,大的值有利于全局搜索,而小的值有利于局部探索. 在优化调度问题中, V_{\max} 被设为 M (任务数量),由于速度可为负值,所以 V_i 是 $[-(M-1), M-1]$ 之间的值, X_{\max} 设定为 M ,由于 X_i 代表任务 ID 号,它只能为正值,所以 X_i 是区间 $[0, M-1]$ 内的值.

(3) 群体规模和最大迭代次数删除英文代号

该参数的设计主要是依据问题的规模. 本问题取群体规模为 20,最大迭代次数为 100.

3.2.2 遗传算法参数设计

结合可重入生产调度优化问题,对各算子的设计进行分析.

(1) 选择算子

选择算子的设计采用比例选择和最优保存的混合策略. 比例选择方对于一个种群大小为 K , 个体适应度为 F_i 的种群, 个体 i 被选中的概率 $P_{is} = F_i^{-1} / \sum_{i=1}^K F_i^{-1}$. 该方法能保证具有最小适应度的个体被选中的概率最大. 而在新种群生成之后,应用最优保存策略,即比较出新旧种群中的最小适应度个体,用该

个体取代新种群中的最大适应度个体.

比例选择方法能以概率选择的方式保证种群的质量和多样性,而最优保存策略则是保证遗传算法收敛的一个重要条件.

(2) 交叉算子

交叉算子的设计和实现与所研究的问题密切相关,一般要求既不要太多地破坏个体编码串中表示优良性状的优良模式,又要能够有效地产生出一些较好的新个体模式. 交叉算子的设计包括两个方面的内容:确定交叉点的位置,基因交换的方法.

本研究采用基本的单点交叉来实现这一算子,其方法为:先随机产生 1 个交叉位置,然后 2 个父染色体以交叉位置为界,互相交换交叉位置前段的染色体,从而生成 2 个新的子染色体. 交叉操作的实施与否由交叉概率 P_c 来决定,一般的取值范围是 0.4~0.9. 本文实验取为 0.7.

(3) 变异算子

变异算子的设计包括两方面的内容:确定变异点的位置,变异位置处的基因替换方法. 在此,设计变异算法如下:随机产生 2 个变异位置,然后交换这 2 个变异位置上的代码值. 变异操作由变异概率 P_m 决定是否进行,一般取为 0.001~0.100. 本文实验设为 0.05.

4 仿真及结果分析

为检验算法的可行性和有效性,采用 Mini-fab 模型作为实例进行仿真计算. 生产加工数据如表 2 所示. 每工件存储费按加工费用的 5% 计算. 适应度函数采用数学规划模型的目标函数,目标函数中的参数 w_1 取为 0.125, w_2 取为 0.875,交货期延迟惩罚因子 P 取为 10.

表2 Mini-fab模型加工参数

Tab.2 Machining parameters of mini-fab model

机器	工步	加工时间/min	加工费用/元
M_1	1	25	37.23
	5	25	43.82
M_2	1	25	24.67
	5	25	56.36
M_3	2	30	21.15
	4	50	16.78
M_4	2	30	15.18
	4	50	18.35
M_5	3	50	32.03
	6	10	12.82

运用混合算法、标准遗传算法和粒子群算法分别对该仿真实例进行了优化求解,对结果进行实验对比.3种算法各进行50次的仿真计算,算法收敛效果如图3所示,并统计得出了表3的实验结果对比,最优结果在混合算法的一次仿真的第172代得到(图4).从结果可以看出,在最优值、最劣值、平均值

以及搜索成功率等几个性能指标方面,混合算法最优,遗传算法其次,粒子群算法最后.在反映收敛速度的性能指标平均成功搜索收敛代数上,粒子群表现最好,其次为混合算法,遗传算法最慢.总体上看,混合算法有着更好的优化性能.

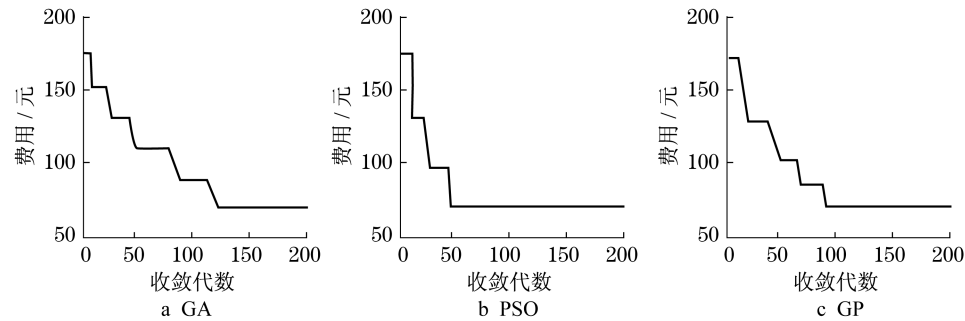
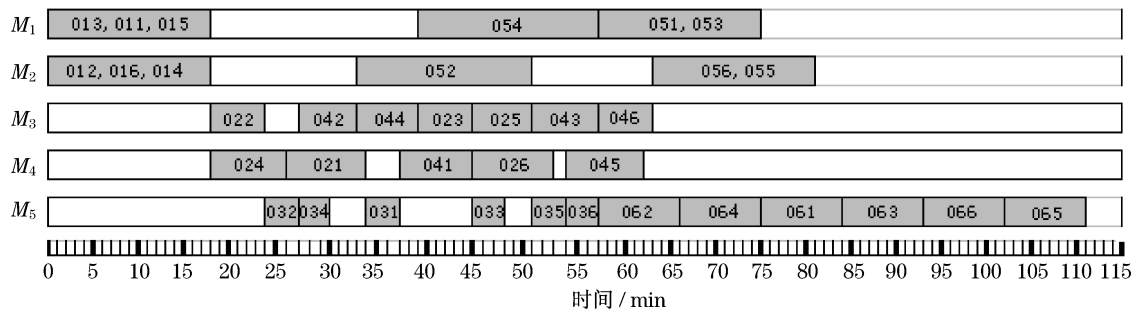


图3 算法收敛图

Fig.3 Comparison of algorithm convergence



注:0ij代表第j卡晶圆的第i道工序.

图4 优化调度方案甘特图

Fig.4 Gantt chart of scheduling optimization

表3 算法结果比较

Tab.3 Comparison of algorithm results

算法	实验结果			搜索成功率/%	平均搜索收敛代数
	最优值	最劣值	平均值		
GA	69.797	172.536	91.781	19.7	117.53
PSO	70.435	177.932	96.293	15.0	41.27
GP	69.212	165.585	85.765	63.0	85.61

5 结语

可重入制造系统的生产调度是一个典型的随机、动态过程^[1].为解决此问题,首先对其进行了系统建模,运用数学模型准确地描述了该调度优化问题,并且将所得的数学模型映射为有向图,便于智能搜索算法的应用.然后,利用粒子群算法和遗传算法的互补性,设计了一种两者结合的混合算法,优化设

置了算法参数,通过与粒子群算法、遗传算法的实例比较,验证了混合算法具有更优的性能,具备了粒子群算法收敛速度快和遗传算法全局搜索能力强的优势,同时很大程度上克服了它们的局限性.但数学规划模型还有待于进一步的完善,相关参数也有待于进一步的优化,这些将作为下一步研究的重点.

参考文献:

- [1] Kumar P R. Re-entrant lines[J]. Queueing Systems, 1993, 13(2):87.
- [2] 王中杰,吴启迪.半导体生产线控制与调度研究[J].计算机集成制造系统,2002,8(8):607.
WANG Zhongjie, WU Qidi. Research of control and scheduling on semiconductor manufacturing line[J]. Computer Integrated Manufacturing Systems, 2002, 8(8):607.

(下转第 772 页)