

控制与决策

Control and Decision

微电子生产过程调度问题基于指标快速预报的分解算法

张龙, 许川佩, 刘民, 董明宇

引用本文:

张龙, 许川佩, 刘民, 等. 微电子生产过程调度问题基于指标快速预报的分解算法[J]. *控制与决策*, 2020, 35(1): 139–146.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2018.0422>

您可能感兴趣的其他文章

Articles you may be interested in

基于总加权完成时间的可重入混合流水车间调度问题

Reentrant hybrid flowshop scheduling problem based on total weighted completion time

控制与决策. 2018, 33(12): 2218–2226 <https://doi.org/10.13195/j.kzyjc.2017.0907>

基于新型教学优化算法的低碳柔性作业车间调度

Novel teaching–learning–based optimization algorithm for low carbon scheduling of flexible job shop

控制与决策. 2017, 32(9): 1621–1627 <https://doi.org/10.13195/j.kzyjc.2016.1008>

求解任务可拆分多项目协同调度问题的启发式算法

Heuristic algorithm for solving multi–project collaborative scheduling problem with activity splitting

控制与决策. 2017, 32(6): 1013–1018 <https://doi.org/10.13195/j.kzyjc.2016.0415>

具有空闲时间的云制造作业车间调度方法

Job shop scheduling method with idle time in cloud manufacturing

控制与决策. 2017, 32(5): 811–816 <https://doi.org/10.13195/j.kzyjc.2016.0447>

面向航空发动机的知识化制造系统拖期调度与自重构

Tardiness scheduling and self–reconfiguration of knowledgeable manufacturing oriented to aircraft engine

控制与决策. 2017, 32(3): 526–534 <https://doi.org/10.13195/j.kzyjc.2016.0179>

集卡分批到达模式下的进口箱场桥作业调度优化

Scheduling optimization of yard crane for import containers based on truck batch arrival pattern

控制与决策. 2016, 31(9): 1654–1662 <https://doi.org/10.13195/j.kzyjc.2015.0875>

基于GMOGSO的多目标流水车间调度问题

Multi–objective flow shop scheduling problem based on GMOGSO

控制与决策. 2016, 31(10): 1772–1778 <https://doi.org/10.13195/j.kzyjc.2015.1309>

改进型细菌觅食算法及多货叉仓库调度应用

Improved bacterial foraging algorithm and application of warehousing scheduling with multi–shuttles

控制与决策. 2015(2): 321–327 <https://doi.org/10.13195/j.kzyjc.2013.1436>

微电子生产过程调度问题基于指标快速预报的分解算法

张 龙^{1,2†}, 许川佩^{1,2}, 刘 氏³, 董明宇³

(1. 桂林电子科技大学 电子工程与自动化学院, 广西 桂林 541004; 2. 广西自动检测技术与仪器重点实验室, 广西 桂林 541004; 3. 清华大学 自动化系, 北京 100084)

摘 要: 微电子生产过程调度问题具有规模大和约束复杂等特点, 如菜单、Setup 时间和组批约束等, 其优化调度具有一定难度. 针对以最小化平均流经时间为调度目标的较大规模微电子生产过程调度问题, 提出一种基于指标快速预报的分解方法 (DM-IFP). 首先, 通过松弛不可中断约束, 设计一种代理方法, 即基于机器负载的操作完工时间快速预测方法 (CTP-ML); 其次, 设计基于 CTP-ML 的问题分解方法, 将原问题迭代分解为多个连续交迭的子问题; 然后, 提出一种基于双信息素的蚁群算法 (ACO-D) 用于求解分解后的子问题, 其全局调度目标采用 CTP-ML 获取, 有效保证了全局优化性能; 最后, 针对一些不同规模的仿真数据, 将所提出方法与一些代表性的算法进行详尽的数值对比, 计算结果表明所提出方法在所获解的质量和收敛性上均有改善.

关键词: 微电子生产过程; 调度; 蚁群算法; 分解; 预测

中图分类号: TP18

文献标志码: A

An indexes fast prediction based decomposition method for scheduling problem in microelectronic production process

ZHANG Long^{1,2†}, XU Chuan-pei^{1,2}, LIU Min³, DONG Ming-yu³

(1. School of Electronic Engineering and Automation, Guilin University of Electronic Technology, Guilin 541004, China; 2. Guangxi Key Laboratory of Automatic Detection Technology and Instrument, Guilin 541004, China; 3. Department of Automation, Tsinghua University, Beijing 100084, China)

Abstract: The scheduling problem in the microelectronic production process has some characteristics including large scale and complex constraints, such as recipe constraint, Setup time, batch capacity and so on. It is difficult to obtain the optimal solution. For the problem with the objective of minimizing the mean cycle time, this paper proposes an indexes fast prediction based decomposition method (DM-IFP). Firstly, after relaxing the non-preemptive constraint, a surrogate method, i.e., the fast prediction method of operation completion time based on the machine load (CTP-ML), is proposed. Then, a CTP-ML based problem decomposition method is designed to decompose the original problems into several consecutive and overlapped subproblems. A double pheromones based ant colony optimization (ACO) algorithm is proposed to solve the subproblem, in which the CTP-ML is applied to obtain the global scheduling objective of each subproblem so that the original scheduling problem is optimized. Finally, based on some simulated data with different scale, sufficient computational comparisons are provided between the proposed DM-IFP and some representative algorithms. It is shown that the proposed method generates better results in terms of quality and convergence.

Keywords: microelectronic production process; scheduling; ant colony optimization; decomposition; Prediction

0 引 言

微电子产业是具有战略资源和国际特征的基础产业, 是信息社会的基石和核心, 对传统产业具有渗透与带动作用, 对国家安全与国防建设起到了关键作用^[1]. 而微电子生产过程调度与优化技术水平一直是微电子产业发展的瓶颈之一. 随着全球市场激烈

的竞争, 微电子生产过程呈现出多品种小批量的特点^[2], 其相应的调度问题规模大 (从投入生产到产品成型往往需要几十道甚至上百道工序) 且约束复杂 (包括菜单约束、Setup 时间约束、组批约束等), 其优化求解具有一定难度, 该研究方向近年来一直受到国内外学术界和工业界的关注.

收稿日期: 2018-04-07; 修回日期: 2018-07-02.

基金项目: 国家自然科学基金项目 (61561012, 61741403); 国家科技重大专项课题 (2011ZX02504-008); 广西高校中青年教师基础能力提升项目 (ky2016YB152); 广西自动检测技术与仪器重点实验室主任基金项目 (YQ16109).

责任编辑: 樊治平.

†通讯作者. E-mail: zhanglong@guet.edu.cn.

针对上述具有大规模特点的离散型生产过程调度问题,国内外学者们试图从问题分解的角度,降低求解规模,提高蚁群算法、遗传算法等群体智能优化算法的求解性能.主要包括基于机器的分解^[3-7]和基于时间的分解方法.机器分解的方法主要是将原调度问题涉及的机器根据一些属性(如瓶颈程度、机器所属流程等)进行分组,对不同的机器组采用不同的策略进行求解,然后协调不同分组之间的约束形成原调度问题的解.然而,由于不同子问题间存在约束,为获得问题的可行解往往需要对子问题反复迭代协调求解^[3],其求解效率往往较低.另外,由于微电子生产过程存在多品种小批量及可重入特性,使得生产过程的瓶颈机器等特性会发生漂移,从而上述基于机器分解的方法难以直接应用于微电子生产过程调度问题.

基于时间的分解方法按加工时间的先后次序对子问题进行顺序求解,子问题间的约束能够被自然满足,不存在子问题反复多次求解的缺陷,该分解方法近年来吸引了广大学者的注意.如Jung等^[8]针对微电子生产过程扩散区调度问题,提出一种基于混合整数线性规划的问题分解方式,本质上是一种滚动时间窗口的分解方法.Guo等^[9-10]针对最小化完工时间为调度目标的微电子生产过程调度问题,采用滚动时域策略对问题进行分解,设计了蚁群算法.上述方法首先通过控制一些参数,如时间窗口长度、子问题包含的操作数等,对子问题的规模进行限制,子问题的调度目标直接采用全局目标;然后,采用或设计数学规划方法、蚁群算法、遗传算法等优化方法进行求解.该方法对子问题后续未调度操作未予考虑,使得子问题的局部优化并不能保证全局调度目标的优化性能.

进而,一些学者提出一些基于预测的问题分解方法.如针对大规模Job Shop调度问题,Liu等^[11]结合滚动时域和工件分解的方法,提出一种基于预测的迭代分解方法,设计了一种RC规则对子问题后续未调度操作安排上机,以预测每个染色体的全局调度目标.Zhang等^[12]采用模拟退火算法优化分解过程,在求解子问题的遗传算法中设计了基于瓶颈操作的免疫算子.Zhai等^[13]结合滚动时域和瓶颈识别的思想,针对大规模Job Shop调度问题,提出一种基于分解的多瓶颈调度算法,采用最小修正交货期优先(MOD)启发式规则调度子问题后续操作,获得每个染色体对应的全局调度目标.在上述基于滚动时域的分解方法中,采用启发式规则在满足问题约束的条件下确定子问题后续未调度操作的开始加工时间,对于每个染色体未调度操作或者通过解码,或者通过启发式规则确定上机顺序,其本质上并未降低问题的求解规模.

本文针对以最小化平均流经时间为调度目标的

较大规模微电子生产过程调度问题,提出一种基于指标快速预报的分解方法(DM-IFP).首先,松弛不可中断约束,设计了一种代理方法,即基于机器负载的操作完工时间快速预测方法(CTP-ML),用于指导原问题迭代分解为多个连续交迭的子问题;在迭代求解过程中,逐步固定子问题中部分操作的开始加工时间,将其余操作滚动到下一迭代子问题.重复上述过程,直到所有Lot均完成调度.然后,提出一种基于双信息素的蚁群算法(ACO-D),用于求解分解后的子问题,其全局调度目标采用CTP-ML快速获取,以有效保证调度问题的全局优化性能.

1 问题描述

微电子生产过程调度问题是一类规模较大、决策类型多的调度问题.以最小化平均流经时间为调度目标的微电子生产过程调度问题可描述如下:

设调度问题包含 I 个Lot和 K 台机器,分别组成Lot集合 $J = \{J_i | i = 1, 2, \dots, I\}$ 和机器集合 $M = \{M_k | k = 1, 2, \dots, K\}$, $r_i \geq 0$ 为Lot J_i 的释放时刻.Lot J_i 由 n_i 个操作 $\{O_{ij} | j = 1, 2, \dots, n_i\}$ 组成, $O = \{O_{ij} | i = 1, 2, \dots, I; j = 1, 2, \dots, n_i\}$ 为调度问题所有操作组成的集合.

在微电子生产过程中, Lot的每一步加工都有一定的处理规范,如加工温度、压力、加工时间等,可用菜单表示^[8].菜单是连接操作和机器的纽带,每个Lot的每个操作均具有某一菜单,而每台机器仅能加工具有某些菜单的操作(以下将机器加工具有某菜单的操作简称为机器加工某菜单).设 $E = \{e_l | l = 1, 2, \dots, L\}$ 为所有菜单的集合,操作 O_{ij} 的菜单为 $e_{ij} \in E$,机器 M_k 可加工的菜单组成集合 $E_k \subset E$,从而当且仅当 $e_{ij} \in E_k$ 时,操作 O_{ij} 可以在机器 M_k 上加工,相应的加工时间 p_{ij}^k 仅由其加工机器和具有的菜单唯一确定.

根据机器的组批能力,所有的机器可分为组批加工机器 $M^b \subset M$ 和串行加工机器 $M^s \subset M$,在任一串行加工机器上,每个时刻仅能加工一个操作,在任一组批加工机器上,每一时刻可加工多个操作,同时加工的操作形成一批,能同时加工的最大操作数称为机器的组批能力,设 B_{lk} 为菜单 $e_l (l = 1, 2, \dots, L)$ 在机器 $M_k (k = 1, 2, \dots, K)$ 上的组批能力.不失一般性,对于串行加工机器 $M_k \in M^s$ 和菜单 $e_l \in E_k$,相应的组批能力 $B_{lk} = 1$.

另外,在机器 $M_k (k = 1, 2, \dots, K)$ 上,若即将加工操作的菜单 e_l 与上一个加工操作的菜单不同,则需要一个额外的Setup时间,令加工操作 O_{ij} 所需的Setup时间标记为 u_{ij} .

本文以最小化平均流经时间为调度目标,设 $c_i \geq 0$ 为 J_i 最后一个操作的完工时间, J_i 的流经时间为

$c_i - r_i$, 则最小化目标函数为 $\sum_{i=1}^I (c_i - r_i) / I$.

考虑到问题的组批约束, 上述微电子生产过程调度问题在每个决策时刻涉及如下4种决策: 1) 机器选择, 从空闲机器中选择某台机器安排操作加工; 2) 菜单选择, 从被选机器待加工的所有操作具有的菜单中选择优先级最高的菜单; 3) 操作选择, 从具有选定菜单的待加工操作中, 选择调度优先级最高的进行加工, 对于串行加工机器, 选择一个具有最高调度优先级的操作进行加工; 4) 组批, 对组批加工机器, 从具有选定菜单的操作中按某种方式组批加工.

2 基于指标快速预报的分解算法(DM-IFP)

所提出的基于指标快速预报的分解算法DM-IFP如图1所示. 在每一阶段, 采用提出的指标快速预报方法CTP-ML快速预报未确定加工开始时间操作的开始及完工时间, 基于此根据给定的子问题包含操作数构造当前子问题; 然后, 采用提出的蚁群算法ACO-D对当前子问题进行求解; 最后, 确定部分操作的加工开始时间, 子问题中其余操作滚动到下一阶段子问题. 上述分解和求解过程迭代进行, 直到获得整个调度问题的解.

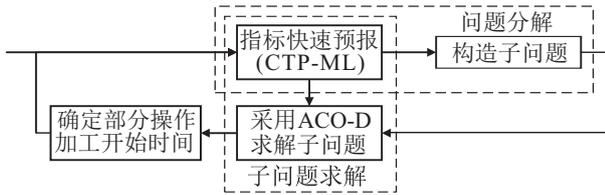


图1 基于指标快速预报的分解算法DM-IFP

2.1 基于机器负载的操作完工时间快速预测方法

为确保迭代分解过程中每个子问题中每个Lot被包含的操作处于合理的调度周期, 及快速获取全局调度目标, 借鉴流模型^[14]思想, 松弛微电子生产过程调度问题中的不可中断约束, 提出一种代理方法, 即一种基于机器负载的操作完工时间的快速预测方法CTP-ML, 以获取未确定加工开始时间操作的近似完工时间.

设 r_{ij} 为操作 O_{ij} 的到达时间, 即操作 O_{ij} 最早可开始加工时间及前继操作 $O_{i,j-1}$ 的完工时间. 操作 O_{ij} 的完工时间 c_{ij} 可根据其可加工机器集合 M_{ij} 前的等待加工操作, 即机器负载, 按如下方法进行预测.

考虑到操作 O_{ij} 在其可加工机器集 M_{ij} 中每台机器 $M_k \in M_{ij}$ 的加工时间 p_{ij}^k 互不相同, 将其在上述所有机器上的平均加工时间 \bar{p}_{ij} 来表征实际加工时间, 即

$$\bar{p}_{ij} = \frac{\sum_{M_k \in M_{ij}} p_{ij}^k}{|M_k \in M_{ij}|}, \quad (1)$$

$|\cdot|$ 表示集合中元素的个数.

进一步, 操作 O_{ij} 平均分配到其可加工机器集合 M_{ij} 中每台机器上的加工负载为

$$\bar{p}_{ij} = \frac{\sum_{M_k \in M_{ij}} p_{ij}^k}{|M_k \in M_{ij}|}. \quad (2)$$

令 W_k 为机器 M_k 前等待加工操作的集合, 机器 M_k 的理论负载可根据下式计算:

$$\Gamma_k = \sum_{O_{ij} \in W_k} \bar{p}_{ij}. \quad (3)$$

根据式(3), 操作 O_{ij} 的完工时间 c_{ij} 可预测为

$$c_{ij} = r_{ij} + \bar{p}_{ij} + \min_{M_k \in M_{ij}} \Gamma_k. \quad (4)$$

进而, Lot J_i 的第 $m (m > j)$ 个操作 O_{im} 的完工时间 c_{im} 可预测为

$$c_{im} = r_{ij} + \sum_{l=j}^m \bar{p}_{il} + \min_{M_k \in M_{il}} \Gamma_k. \quad (5)$$

操作 O_{im} 的到达时间可预测为

$$\bar{r}_{im} = \bar{c}_{i,m-1}, \quad (6)$$

Lot J_i 的完工时间 c_i 可根据其第 j 个操作的到达时间 r_{ij} 及各机器的理论负载 Γ_k 通过下式预测:

$$c_i = r_{ij} + \sum_{m=j}^{n_i} \bar{p}_{im} + \min_{M_k \in M_{im}} \Gamma_k. \quad (7)$$

2.2 基于操作到达时间预测的问题迭代分解

本文采用滚动时域的方法, 通过限定子问题包含的操作数, 对调度问题进行顺序迭代搭接式分解.

2.2.1 子问题的构造

对以最小化平均流经时间为调度目标的微电子生产过程调度问题, 根据Lot的工艺路径、时间窗口及操作的预测到达时间, 从未调度操作中选取当前可加工及部分将来可加工的操作构成子问题, 子问题的调度目标为全局调度目标的预测值. 设当前阶段(第 k 阶段)已调度操作(即已确定加工开始时间的操作)组成集合 B_k , $b_i (i = 1, 2, \dots, I)$ 为工件 J_i 中属于集合 B_k 的最大操作号, 即

$$b_i = \max\{j | O_{ij} \in B_k\}. \quad (8)$$

然后, 根据式(5)和(6)对不属于操作集 B_k 的操作 $O_{ij} (j > b_i)$ 的到达时间 r_{ij} 进行预测, 并从中选取 N_s 个 \bar{r}_{ij} 最小的操作组成子问题的操作集 H_k , 其余操作形成操作集合 P_k . 根据上述子问题的形成过程, 第 k 阶段, 子问题的数学规划模型可描述为

$$\min f = \sum_{i=1}^I \frac{\bar{c}_i - r_i}{I}. \quad (9)$$

s.t.

$$s_{ij} - s_{i,j-1} \geq p_{i,j-1}^{k(i,j-1)} + u_{ij}, \quad \forall O_{ij} \in H_k; \quad (10)$$

$$c_{ij} - s_{ij} = p_{ij}^{k(i,j)}, \quad \forall O_{ij} \in H_k; \quad (11)$$

$$\begin{aligned} \omega_{mn,ij}^k(p_{mn}^k + u_{ij}) - N\omega_{ij,mn}^k(1 - \omega_{mn,ij}^k) \leq \\ s_{ij}^k - s_{mn}^k \leq N\omega_{mn,ij}^k(1 - \omega_{ij,mn}^k) - \omega_{ij,mn}^k(p_{ij}^k + u_{mn}), \\ \forall O_{ij}, O_{mn} \in H_k, m \neq i, n \neq j; \end{aligned} \quad (12)$$

$$\sum_{l \in E_k} \sigma_{lk}^t = 1, \forall k, t \geq \max\{r_{ij} | O_{ij} \in B_k\}; \quad (13)$$

$$\sum_{k=1}^K \tau_{ik}^t = 1, \forall i, t \geq \max\{r_{ij} | O_{ij} \in B_k\}; \quad (14)$$

$$\sum_{i=1}^N \tau_{ik}^t \leq \sum_{l=1}^L \sigma_{lk}^t B_{lk}, \forall k, t \geq \max\{r_{ij} | O_{ij} \in B_k\}; \quad (15)$$

$$s_{i,b_i+1}^{k(i,b_i+1)} \geq c_{i,b_i+1}, \forall i \leq I. \quad (16)$$

式(9)为子问题的目标函数:最小化全局预测的平均流经时间, \bar{c}_i 为 J_i 的预测完工时间,通过式(7)获得.式(10)给出了工艺路径约束和 Setup 时间约束, $k(i, j-1)$ 表示 $O_{i,j-1}$ 的加工机器;式(11)给出了不可中断约束;式(12)给出了在同一台机器上两个操作的先后加工顺序约束,若在机器 M_k 上操作 O_{ij} 先于操作 O_{mn} 加工,则 $\omega_{ij,mn}^k = 1$, 否则, $\omega_{ij,mn}^k = 0$; N 是一极大的正数;式(13)表明任一机器同一时刻只能加工一个菜单,若时刻 t 在机器 M_k 上加工的菜单为 e_l , 则 $\sigma_{lk}^t = 1$, 否则, $\sigma_{lk}^t = 0$;式(14)表明一个操作只能在一台机器上加工,若时刻 t , Lot J_i 在机器 M_k 上加工,则 $\tau_{ik}^t = 1$, 否则, $\tau_{ik}^t = 0$;式(15)给出了组批约束;式(16)给出子问题中每个 Lot 第 1 个操作的开始时间受已确定加工开始时间操作集 B_k 中相应操作完工时间的约束.

2.2.2 子问题的调整

所提出分解算法需从一个阶段向下一个阶段推进,直到调度问题所有操作的开始加工时间均被确定.在子问题滚动推进过程,采用子问题搭接的方式处理两个连续子问题关系,具有如下优点:1)由于每个子问题考虑了更多的操作,从而算法的整体优化效果会更好;2)子问题包含的 Lot 的开始时间大致相同;3)可以将每台组批加工机器最后一批组批规模小于最大组批能力的批次放入下一阶段子问题求解,提高组批效率.

基于上述分析,设搭接操作集 R_k 规模为 N_r , 子问题中已安排操作加工的组批加工机器集合为 $\Omega^b \subseteq M^b$, 可采用如下流程从当前阶段子问题中选择部分操作作为搭接操作移入下一阶段子问题进行求解.

Step 1: 令 $R_k = \emptyset$.

Step 2: 对于每一台机器 $M_i \in \Omega^b$, 若最后一批加工的操作所属的批次小于最大组批能力,则该批所有操作均为搭接操作,即将该批所有操作并入操作集

R_k 中.

Step 3: 若 R_k 中所包含操作的总数小于 N_r , 则执行如下步骤,否则,结束流程:

Step 3.1: 从 H_k 中顺序选择部分加工开始时间 s_{ij} 最大的操作放入操作集 R_k 中,使得操作集 R_k 规模达到 N_r ;

Step 3.2: 设 Step 3.1 移入操作集 R_k 中所有的操作中,最小的加工开始时间对应的操作为 $O_{i^*j^*}$, 则将与操作 $O_{i^*j^*}$ 同批的所有操作移入到操作集 R_k .

3 基于双信息素的蚁群算法

蚁群算法与遗传算法、粒子群算法等群体智能优化方法相比,蚁群算法具有较强的鲁棒性、优良的分分布式计算机制、易于与其他算法结合等优点^[15-16].同时,蚁群算法是一种基于模型的优化算法^[9],对问题实例的复杂约束、规模大等特点敏感性不大.近年来,一些学者已将蚁群算法用于求解微电子生产过程调度问题^[9],取得了较好的调度效果,但其所求解的微电子生产过程调度问题,对实际问题作了较大的简化,特别是,忽略了问题中的菜单约束.

本文针对具有菜单、Setup 时间和组批约束的微电子生产过程调度问题,给出一种基于双信息素的蚁群算法(ACO-D),用于对问题分解后的子问题进行求解,菜单信息素用于菜单的选择,操作信息素用于菜单约束下操作的选择.下面将分别给出菜单和操作的状态转移概率,及信息素更新机制,并为有效改善由于正反馈机制造成的蚁群算法早熟现象,给出一种算法重新启动机制.

3.1 状态转移概率

为有效处理菜单约束,针对菜单选择决策和操作排序决策,以下分别给出菜单选择概率和操作选择概率.其中,对于选定的机器,设时刻 t 可加工的操作具有的菜单集为 $E^t = \{e_1, e_2, \dots, e_{L_t}\}$, 则菜单 e_l 的选择概率

$$\text{Pr}_l^t = \begin{cases} \frac{\left(\sum_{e_{l'} \in E^t, l' \neq l} \tau_{l,l'} \right)^\alpha (\eta_l)^\beta}{\sum_{e_{l''} \in E^t} \left(\sum_{e_{l'} \in E^t, l' \neq l''} \tau_{l'',l'} \right)^\alpha (\eta_{l''})^\beta}, & e_l \in E^t; \\ 0, & \text{else.} \end{cases} \quad (17)$$

其中: $\tau_{l,l'}$ 为菜单 e_l 优先于菜单 $e_{l'}$ 的信息素浓度; η_l 为启发式信息,采用在当前机器待加工且具有菜单 e_l 的操作的 DRLB^[17] 平均值表示; α 为信息素因子, β 为启发式因子.

对于选定的菜单 e_{l^*} , 设 t 时刻可加工的集为 $O^t = \{O_1, O_2, \dots, O_{K_t}\}$, 操作 O_i 的选择概率为

$$Pr_i^o = \begin{cases} \frac{\left(\min_{O_j \in O^t, O_j \neq O_i} \tau_{ij}\right)^\alpha (\eta_i)^\beta}{\sum_{O_m \in O^t} \left(\min_{O_j \in O^t, O_j \neq O_m} \tau_{mj}\right)^\alpha (\eta_m)^\beta}, & O_i \in O^t; \\ 0, & \text{else.} \end{cases} \quad (18)$$

其中: τ_{ij} 为操作 O_i 优先于操作 O_j 的信息素浓度; η_i 为操作 O_i 启发式信息, 用来引导蚁群算法路径的生成, 采用 DRLB^[17] 给出的调度优先级指标表示.

3.2 操作选择策略

基于上述信息素, t 时刻的操作选择策略可以分为如下几类:

1) 被选机器 M_{k^*} 为串行加工机器.

Step 1: 菜单选择, 根据式(17)给出的菜单选择概率, 按照轮盘赌的方式从 E^t 选出加工菜单 e_{l^*} ;

Step 2: 操作选择, 根据式(18)给出的操作选择概率, 按照轮盘赌的方式从 O^t 中选出在机器 M_{k^*} 上加工的操作.

2) 被选机器 M_{k^*} 为组批加工机器.

从 E^t 中选出操作数大于或等于机器 M_{k^*} 组批能力的菜单组成菜单集 $E_s^* = \{e_l | n_l \geq B_{lk^*}\}$, 其中 n_l 为 M_{k^*} 待加工且具有菜单 e_l 的操作总数, B_{lk^*} 为 M_{k^*} 的组批能力, 即能同时加工属于菜单 e_l 的最大操作总数.

① 若 $E_s^* \neq \emptyset$, 则:

Step 1: 菜单选择, 根据式(17)给出的菜单选择概率, 按照轮盘赌的方式从 E_s^* 中选出加工菜单 e_{l^*} ;

Step 2: 操作选择, 根据式(18)给出的操作选择概率, 按照轮盘赌的方式从 O^t 中顺序选出 $B_{l^*k^*}$ 个操作组成一批在机器 M_{k^*} 上加工.

② 若 $E_s^* = \emptyset$, 则根据 MBS 组批规则^[18], 从 E^t 中选出操作数大于最小组批规模的菜单组成菜单集 $E_m^* = \{e_l | n_l \geq B_{lk^*}^{\min}\}$, $B_{lk^*}^{\min}$ 为 M_{k^*} 的最小组批规模, 即能同时加工属于菜单 e_l 的最小操作总数.

i) 若 $E_m^* \neq \emptyset$, 则:

Step 1: 菜单选择, 根据式(17)给出的菜单选择概率, 按照轮盘赌的方式从 E_m^* 选出加工菜单 e_{l^*} ;

Step 2: 操作选择, 选择具有菜单 e_{l^*} 的所有待加工操作组成一批在机器 M_{k^*} 上加工.

ii) 若 $E_m^* = \emptyset$, 则 M_{k^*} 空闲.

下面给出一个实例说明上述决策过程. 选定空闲机器前待加工操作属性如表1所示, 包含7个操作, 分属于3个不同的菜单; 表2给出了3个菜单的信息素, 如0.5表示菜单1 优先菜单2加工的信息素 τ_{12} ; 操作信息素如表3所示, 从中可以看出, 由于菜单约束

的存在, 具有不同菜单操作间的信息素为0, 简化了操作选择过程.

表1 操作属性

操作	1	2	3	4	5	6	7
菜单	1	1	1	2	2	3	3
DRLB	1	2	3	1	2	2	3

表2 菜单信息素

菜单	1	2	3
1	0	0.5	0.6
2	0.3	0	0.3
3	0.2	0.7	0

表3 操作信息素

操作	1	2	3	4	5	6	7
1	0	0.5	0.6	0	0	0	0
2	0.5	0	0.3	0	0	0	0
3	0.3	0.7	0	0	0	0	0
4	0	0	0	0	0.6	0	0
5	0	0	0	0.3	0	0	0
6	0	0	0	0	0	0	0.2
7	0	0	0	0	0	0.6	0

若 $\alpha = 2, \beta = 3$, 则根据式(17), 菜单1的选择概率计算如下:

$$\begin{aligned} & \sum_{e_{l''} \in E^t} \left(\sum_{e_{l'} \in E^t, l' \neq l''} \tau_{l''l'} \right)^\alpha (\eta_{l''})^\beta = \\ & (0.5 + 0.6)^2 \times \left(\frac{1 + 2 + 3}{3} \right)^3 + \\ & (0.3 + 0.3)^2 \times \left(\frac{1 + 2}{2} \right)^3 + \\ & (0.2 + 0.7)^2 \times \left(\frac{2 + 3}{2} \right)^3 = 23.6, \\ Pr_1^r &= (0.5 + 0.6)^2 \times \left(\frac{1 + 2 + 3}{3} \right)^3 \div 23.6 = 0.41. \end{aligned}$$

相似地, 可以计算得 $Pr_2^r = 0.05, Pr_3^r = 0.54$.

操作1的选择概率可计算如下:

$$\begin{aligned} & \sum_{O_m \in O^t} \left(\min_{O_j \in O^t, O_j \neq O_m} \tau_{mj} \right)^\alpha (\eta_m)^\beta = \\ & (\min\{0.5, 0.6\})^2 \times 1^3 + (\min\{0.5, 0.3\})^2 \times 2^3 + \\ & (\min\{0.3, 0.7\})^2 \times 3^3 = 3.4, \\ Pr_1^o &= (\min\{0.5, 0.6\})^2 \times 1^3 \div 3.4 = 0.07. \end{aligned}$$

相似地, 可计算得到 $Pr_2^o = 0.21, Pr_3^o = 0.72, Pr_4^o = 0.33, Pr_5^o = 0.67, Pr_6^o = 0.03, Pr_7^o = 0.97$.

基于上述选择概率, 操作选择可按如下流程决策:

if M_{k^*} 为串行加工机器

菜单选择: 若产生的随机数为0.2, 则根据轮盘赌方法选择菜单1加工;

操作选择: 若产生的随机数为0.2, 则根据轮盘赌方法选择操作2加工.

else if M_{k^*} 上3个菜单的组批能力均为2, $E_s^* = \{1, 2, 3\}$;

菜单选择:若产生的随机数为0.2,则根据轮盘赌方法选择菜单1加工;

操作选择:若产生的随机数序列为0.2和0.3,则选择操作2和操作3组批加工.

else if M_{k^*} 上3个菜单的组批能力均为5, $E_s^* = \emptyset$

if最小组批规模均为2, $E_m^* = \{1, 2, 3\}$

菜单选择:若产生随机数为0.2,则根据轮盘赌方法选择菜单1加工;

操作选择:选择操作1、操作2和操作3组批加工.

else if最小组批规模均为4, $E_m^* = \emptyset$

M_{k^*} 空闲等待.

3.3 信息素更新及初始化

在蚁群算法每次迭代完成后,设 S_{l_s} 和 S_{g_s} 分别为每次迭代的最优解和迄今为止的全局最优解,每次迭代后,若 S_{l_s} 优于 S_{g_s} ,则 S_{g_s} 被 S_{l_s} 替换.分别采用下式对信息素 $\tau_{l,v}$ 和 τ_{ij} 进行更新:

$$\tau_{l,v} = (1 - \rho)\tau_{l,v} + \rho\Delta\tau^b, \quad (19)$$

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho\Delta\tau^b. \quad (20)$$

其中 $\Delta\tau^b = 1/f_{\text{best}}$, f_{best} 为全局最优解 S_{g_s} 或当前迭代最优解 S_{l_s} 对应的目标函数值.在算法初始迭代阶段采用当前迭代最优解 S_{l_s} 对应的目标函数值更新信息素,以加强算法的局部搜索能力,而在后续迭代阶段,若连续 T_{\min} 代全局最优解 S_{g_s} 未被更新,则用全局最优解 S_{g_s} 代替当前迭代最优解 S_{l_s} 对信息素进行更新,以加强算法的全局搜索能力,使算法能从局部最优跳跃出来.

每一个子问题开始求解时,菜单信息素 $\tau_{l,v}$ 直接继承上一阶段子问题求解结束时的菜单信息素,以加速算法的收敛;而操作信息素 τ_{ij} 在每个子问题求解前均需要重新初始化.

3.4 重新启动机制

上述用全局最优解 S_{g_s} 代替当前迭代最优解 S_{l_s} 对信息素进行更新后,若连续 T_{\min} 代全局最优解 S_{g_s} 仍未被更新,则蚁群搜索算法有可能陷入局部极小.为使得算法从局部极小跳出,重新初始化算法.

4 数值计算及分析

4.1 仿真实例及算法参数设置

本文模拟某微电子企业的实际生产数据形成不同规模的调度问题实例进行数值计算.实际数据中包含70个菜单和117台机器,其中77台机器为组批加工机器,其余的为串行加工机器.组批机

器中,最大和最小的组批能力分别为8和2.根据上述实际数据,随机生成了5类微电子生产过程调度问题实例,分别包含500,1000,2000,3000和5000个Lot,每类包含10个问题实例,分别命名为I1_1, I1_2, ..., I1_10, I2_1, I2_2, ..., I2_10, ..., I5_10.

所提出基于预测分解的蚁群算法(简称DM-IFP)的主要参数通过多次实验对比及参考相关文献设置如下:子问题包含操作数 N_s 为100;搭接操作集 R_k 规模 N_r 为10;ACO-D最大迭代次数设置为100;ACO-D最大重新启动次数为10;触发重新启动的 T_{\min} 为3;蚁群规模为50;参考D-CACO的数值仿真结果^[9],ACO-D中信息素因子和启发式因子 α 和 β 分别设为2和3,遗忘因子 ρ 设置为0.1.

4.2 对比算法

本文选择如下一些求解微电子生产过程调度问题或相似问题的具有代表性算法作为对比算法,以验证DM-IFP的有效性.

1) 蚁群算法(D-CACO)^[9]:在其讨论的问题中,菜单约束未被考虑,且缺少指标预报环节,计算效率难以得到保证.主要操作算子及参数设置为^[9]:

① 在每个子问题求解过程中,对后续操作采用FCFS,即先到先加工的规则进行调度以获得全局目标;

② 算法的终止准则修改为最大计算时间,即每次迭代后判断其计算时间是否超过DM-IFP的计算时间,若超过则终止算法求解;

③ 其他主要算法参数与DM-IFP设置相同.

2) 遗传算法(GA)^[19].

① 为保证菜单约束被满足,修改解的表示为菜单排列表,选择方法设置为轮盘赌,部分匹配交叉(Partial-mapped crossover, PMX)用于交叉操作,逆变异用作变异操作,终止准则如同D-CACO;

② 种群规模为Lot总数的一半,复制概率为0.2,交叉概率为0.95,变异概率为0.05.

3) 模拟退火算法(SA)^[20].

① 与GA相似,修改解的表示为菜单排列表,逆变异和交换变异用于生成调度解的邻域,终止准则修改为与D-CACO相同;

② 初始温度为随机产生500个解中最好解与最差解目标函数之差,冷却速率设为0.995,在每一个温度的邻域搜索次数为200.

4.3 计算结果

表4~表8分别给出了5类不同规模调度问题实例4种迭代优化算法的数值计算结果.

表4 具有500个Lot的调度问题实例

	I1_1	I1_2	I1_3	I1_4	I1_5	I1_6	I1_7	I1_8	I1_9	I1_10
GA	17.75	17.06	17.59	16.26	18.13	17.35	17.31	17.32	17.33	16.35
D-CACO	18.77	17.37	18.36	16.54	18.71	17.43	17.77	18.12	17.56	16.9
SA	17.18	16.69	17.74	16.08	17.72	16.9	16.75	16.62	16.93	16.42
DM-IFP	16.89	16.47	16.43	14.29	17.51	16.25	16.6	16.38	16.71	15.55

表5 具有1000个Lot的调度问题实例

	I2_1	I2_2	I2_3	I2_4	I2_5	I2_6	I2_7	I2_8	I2_9	I2_10
GA	27.04	26.76	25.71	26.91	26.18	27.23	28.39	27.31	26.6	27.53
D-CACO	27.15	27.32	26.11	26.51	28.2	27.64	28.3	28.44	26.31	28.64
SA	25.59	25.44	25.81	25.52	25.31	25.93	26.9	27.29	25.84	26.6
DM-IFP	25.65	25.51	24.52	25.47	24.8	25.7	26.79	26.8	24.2	25.65

表6 具有2000个Lot的调度问题实例

	I3_1	I3_2	I3_3	I3_4	I3_5	I3_6	I3_7	I3_8	I3_9	I3_10
GA	46.7	46.21	46.7	45.36	43.75	45.24	44.71	47.38	46.53	46.99
D-CACO	45.14	45.32	45.8	46.37	42.14	44.47	43.39	45.27	47.63	47.7
SA	45.89	46.43	43.37	43.96	42.71	43.55	44.09	44.81	48.01	45.25
DM-IFP	44.7	44.72	43.18	43.71	42.3	42.64	42.5	44.8	44.17	45.35

表7 具有3000个Lot的调度问题实例

	I4_1	I4_2	I4_3	I4_4	I4_5	I4_6	I4_7	I4_8	I4_9	I4_10
GA	64.98	63.91	63.26	62.93	63.7	65.04	66.66	63.73	65	64.5
D-CACO	64.6	61.8	64.67	64.21	65.9	65.01	66.3	63.18	64.66	65.18
SA	65.82	59.51	65.16	63.94	66.92	68.1	64.22	66.4	64.13	62.13
DM-IFP	63.91	59.36	61.46	60.18	61.42	63.33	62.72	61.36	61.72	63.22

表8 具有5000个Lot的调度问题实例

	I5_1	I5_2	I5_3	I5_4	I5_5	I5_6	I5_7	I5_8	I5_9	I5_10
GA	99.13	100.5	102.7	98.98	101.5	99.67	100.4	101.9	101.4	102.4
D-CACO	99.1	99.34	104.2	99.18	102.2	101.1	102.4	101.8	102.1	102.1
SA	104.1	98.81	101.6	96.19	106.5	103.8	102.6	97.79	98.91	98.08
DM-IFP	97.4	97.12	99.71	97.09	98.2	96.18	97.8	98.17	97.2	97.16

数值计算结果显示:相对于其他迭代优化算法,DM-IFP更适合于本文所研究的调度问题.与其他算法相比,50个问题实例中,在相近的求解时间内有43个实例DM-IFP获得了最好解;相对于GA、D-CACO和SA,采用DM-IFP获得的平均流经时间的平均改进百分比分别为4.2%,4.63%,3.31%.

模拟退火算法是基于单个体的局部搜索算法,在相同的求解时间内有6个问题实例获得了最好解,而基于种群的迭代搜索算法GA,由于问题规模大,求解效率低,有限的求解时间内没有任何问题实例获得最好解.而D-CACO在子问题求解过程中,没有考虑全局目标,局部最优并不能保证全局最优,只有1个问题实例获得了最好解.

图2给出了4种算法求解问题实例I3_1的收敛曲线,横坐标为DM-IFP的子问题序号.对于其他3种算法而言,取DM-IFP子问题求解时间对应的目标函数值,从中可以看出DM-IFP具有更好的收敛速度.随着时间的推进指标预报越来越准确,而在部分子问题求解中,目标函数会有波动,这是由于约束松弛导致指标预报存在一定偏差.

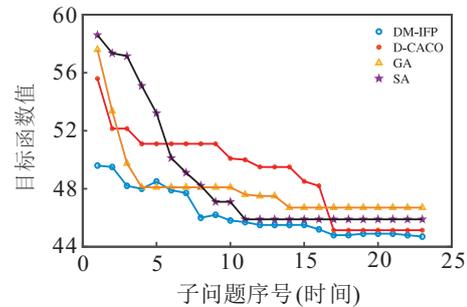


图2 4种对比算法的收敛曲线

总之,本文给出的DM-IFP在求解效率和获得最优解的概率方面,相对于其他相似算法和传统迭代搜索算法均有较大提升.

5 结论

本文针对以最小化平均流经时间为调度目标,具有菜单、Setup时间和组批约束的较大规模微电子生产过程调度问题,提出了一种基于指标快速预报的分解方法(DM-IFP).首先,通过松弛不可中断约束,对未确定加工时间操作的完工时间快速预测,以合理构造当前迭代阶段的子问题.然后,考虑所研究问题的特殊约束,提出一种基于双信息素的蚁群算法(ACO-

D)用于求解分解后的子问题,其全局调度目标采用CTP-ML获取,将局部调度目标与全局调度目标相关联,局部子问题的优化在一定程度上保证了全局调度目标的优化;进一步,为提高所提出算法的优化性能,每一迭代阶段子问题求解后,只固定部分操作的开始加工时间,将其余操作滚动到下一迭代子问题。

对于本文所涉及的问题,进一步的研究方向为如何通过迭代搜索过程中表现出来的统计特性,获取影响调度目标的关键问题特征,有效改进求解算法的性能。

参考文献(References)

- [1] 张兴,黄如,刘晓彦.微电子学概论[M].第3版.北京:北京大学出版社,2010:1-20.
(Zhang X, Huang R, Liu X Y. Introduction to microelectronics[M]. 3rd ed. Beijing: Peking University Press, 2010: 1-20.)
- [2] Yang F, Gao K, Simon I W, et al. Decomposition methods for manufacturing system scheduling: A survey[J]. IEEE/CAA J of Automatica Sinica, 2018, 5(2): 389-400.
- [3] Zhai Y, Sun S, Wang J, et al. A heuristic algorithm for large-scale job shop scheduling based on operation decomposition using bottleneck machine[C]. 2010 Int Conf on Management and Service Science. Wuhan, 2010: 1-4.
- [4] 左燕,谷寒雨,席裕庚.大规模流水线调度的瓶颈分解算法研究[J].控制与决策,2006,21(4):425-429.
(Zuo Y, Gu H Y, Xi Y G. Bottleneck-based decomposition algorithm for large-scale flow shop scheduling problems[J]. Control and Decision, 2006, 21(4): 425-429.)
- [5] Adams J, Balas E, Zawack D. The shifting bottleneck procedure for job shop scheduling[J]. Management Science, 1988, 34(3): 391-401.
- [6] Toscano A, Ferreira D, Morabito R. A decomposition heuristic to solve the two-stage lot sizing and scheduling problem with temporal cleaning[J]. Flexible Services and Manufacturing J, 2019, 31(1): 142-173.
- [7] Wang K, Choi S H. A decomposition-based approach to flexible flow shop scheduling under stochastic setup times[C]. The 5th IEEE Int Conf Intelligent Systems. London: IEEE, 2010: 55-60.
- [8] Jung C, Pabst D, Ham M, et al. An effective problem decomposition method for scheduling of diffusion processes based on mixed integer linear programming[J]. IEEE Trans on Semiconductor Manufacturing, 2014, 27(3): 357-363.
- [9] Guo C T, Jiang Z B, Zhang H, et al. Decomposition-based classified ant colony optimization algorithm for scheduling semiconductor wafer fabrication system[J]. Computers & Industrial Engineering, 2012, 62(1): 141-151.
- [10] 郭乘涛,江志斌,张怀.基于问题分解的蚁群算法在半导体晶圆制造调度中的应用[J].上海交通大学学报,2009,43(11):1798-1802.
(Guo C T, Jiang Z B, Zhang H. Decomposition based ant colony optimization algorithm applied to semiconductor wafer fabrication system[J]. J of Shanghai Jiaotong University, 2009, 43(11): 1798-1802.)
- [11] Liu M, Hao J H, Wu C. A prediction based iterative decomposition algorithm for scheduling large-scale job shops[J]. Mathematical and Computer Modelling, 2008, 47(3/4): 411-421.
- [12] Zhang R, Wu C. A hybrid approach to large-scale job shop scheduling[J]. Applied Intelligence, 2010, 32(1): 47-59.
- [13] Zhai Y, Liu C, Chu W, et al. A decomposition heuristics based on multi-bottleneck machines for large-scale job shop scheduling problems[J]. J of Industrial Engineering and Management, 2014, 7(5): 1397-1414.
- [14] Bertsimas D, Gamarnik D. Asymptotically optimal algorithms for job shop scheduling and packet routing[J]. J of Algorithms, 1999, 33(2): 296-318.
- [15] Dorigo M. Optimization, learning and natural algorithms[D]. Milano: Politecnico di Milano, 1992.
- [16] 李元振,潘全科,李俊青.基于蚁群算法的逻辑函数化简[J].计算机工程与设计,2007,28(12):2788-2789.
(Li Y Z, Pan Q K, Li J Q. Implementation of logic function simplification using ant algorithm[J]. Computer Engineering and Design, 2007, 28(12): 2788-2789.)
- [17] Zhang L, Liu M, Hao J H, et al. Scheduling semiconductor wafer fabrication using a new harmony search algorithm based on receipt priority interval[J]. Chinese J of Electronics, 2016, 25(5): 866-872.
- [18] Neuts M F. A general class of bulk queues with poisson input[J]. Annals of Mathematical Statistics, 1967, 38(3): 759-770.
- [19] Noroozi A, Mokhtari H, Kamal Abadi I N. Research on computational intelligence algorithms with adaptive learning approach for scheduling problems with batch processing machines[J]. Neurocomputing, 2013, 101: 190-203.
- [20] Chou F, Wang H, Chang P. A simulated annealing approach with probability matrix for semiconductor dynamic scheduling problem[J]. Expert Systems with Applications, 2008, 35(4): 1889-1898.

作者简介

张龙(1975—),男,副教授,博士,从事生产过程智能优化与调度等研究,E-mail: zhanglong@guet.edu.cn;

许川佩(1968—),女,教授,博士,从事测控技术与仪器等研究,E-mail: xcp@guet.edu.cn;

刘民(1965—),男,教授,博士,从事生产过程智能优化与调度等研究,E-mail: lium@tsinghua.edu.cn;

董明宇(1978—),男,副研究员,从事智能制造、药物智能设计等研究,E-mail: dmy@mail.tsinghua.edu.cn.

(责任编辑:孙艺红)