

DOI : 10.3901/JME.2014.18.203

基于设备驱动和实质路径的动态并行综合 柔性调度算法*

谢志强^{1,2} 桂忠艳¹ 杨 静²

(1. 哈尔滨理工大学计算机学院 哈尔滨 150080 ;
2. 哈尔滨工程大学计算机学院 哈尔滨 150001)

摘要 : 针对静态重叠时间段在实际加工过程中并非一定重叠, 导致工序并行处理效果不佳的问题, 提出基于设备驱动和实质路径的复杂单产品动态并行综合柔性调度算法。该算法先按最短加工时间为工序选择计划加工设备, 初步缩小工序加工设备的选择范围; 再采用空闲设备驱动策略动态确定并行工序, 对并行工序按提出的并行优化分配策略和最早加工结束策略确定并行加工设备。在采用并行优化分配策略时, 运用实质短路径策略, 避免了某空闲设备上存在多个只能串行加工的工序时, 出现较长实质路径的情况。由于该算法在每一个设备驱动时刻, 动态的分配可重叠加工的工序到不同的设备上同时加工, 实现工序间实质性的并行处理。实例表明: 该算法在不提高算法复杂度的前提下, 不仅兼顾设备驱动策略和动态实质短路径策略的优点, 而且提高了复杂单产品综合柔性调度设备利用率和并行处理率, 缩短产品的完成时间。

关键词 : 综合柔性调度; 设备驱动; 并行优化分配; 最早加工结束; 动态实质短路径

中图分类号: TP278

Dynamic Parallel Integrated Flexible Scheduling Algorithm Based on Device Driver and Essential Path

XIE Zhiqiang^{1,2} GUI Zhongyan¹ YANG Jing²

(1. College of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080;
2. College of Computer Science and Technology, Harbin Engineering University, Harbin 150001)

Abstract : Aiming at the problem that the static overlap time period may not necessarily be overlapped in practical processing process, which may lead the effect of parallel processing not ideal, a dynamic parallel integrated flexible scheduling algorithm based on device driver and essential path has been proposed. The algorithm firstly selects plan processing device for procedures according to short time strategy to preliminary narrow down the selection scope of procedures' processing devices. Idle device driver strategy has been adopted to confirm procedures which can be parallel processed, and the parallel optimization allocation strategy and the earliest processing end strategy have been proposed to confirm processing devices for these parallel processed procedures; In the parallel optimization allocation strategy, when a free machine has more than one schedulable procedure and all of these schedulable procedures can only serial process on this machine, scheduling the procedure according to essential short path strategy in order to avoid the appearance of much longer essential path on these machines. As this algorithm dynamically allocate procedures which can be overlapped processed to different machines to process at each device driver moment, so it can realize full parallel processing of procedures. Examples show that the proposed algorithm absorbs the merits of both device driver strategy and dynamic essential short path strategy under the premise of not increasing the algorithm's complexity, the proposed algorithm also improves integrated flexible scheduling equipments utilization ratio and the parallel processing rate of complex single product, the total product processing time has been reduced.

Key words : integrated flexible scheduling ; device driver ; parallel optimization allocation ; earliest processing end ; dynamic essential short path

0 前言

柔性车间调度 (Flexible job-shop scheduling

problem, FJSP) 是对传统车间调度问题的扩展。近年来对产品柔性制造调度的研究已经取得了一定的成果, 如张超勇等^[1]针对柔性车间调度问题的特点, 设计了基于工序编码和基于机器分配编码的交叉和变异算子, 提出一种双层子代产生模式的改进遗传算法; XING 等^[2]提出基于蚁群与邻域搜索算法的混合算法, 以生产周期、设备总负荷和关键设备负荷为目标, 对标准柔性作业调度问题进行了研究;

* 国家自然科学基金(60873019, 61370086, 61370083)、黑龙江省博士后科研启动(LBH-Q13092)、黑龙江省自然科学基金(F201101)、黑龙江省教育厅科技(12531105)和现代制造技术与刀具开发黑龙江省高校重点实验室开放课题(2012DJ-003)资助项目。20130907 收到初稿, 20140506 收到修改稿

XIE 等^[3]针对柔性车间调度问题提出了一个新的结合知识策略的遗传算法;GHASEM 等^[4]提出了基于粒子群和局部搜索算法的集成优化方法,以最小化完工时间、机器总负荷和机器最大负荷为目标,对含不同释放时间的柔性车间调度问题进行研究;施进发等^[5]提出了交货期惩罚下柔性车间调度多目标 Pareto 优化算法,对柔性车间作业进行多变量、多约束下的调度布局优化设计,采用约束 Pareto 支配混合禁忌搜索算法,减小邻域搜索空间;张国辉等^[6]提出改进遗传算法求解柔性车间作业调度问题,考虑各机器负荷平衡,设计一种全局搜索、局部搜索和随机产生相结合的初始化方法,加快遗传算法的收敛速度;鞠全勇等^[7]提出了多种群粒子群与遗传算法相结合的算法,以生产周期、最大提前/拖后时间、生产成本和设备利用率为目标,对包含机床工时费、工人小时工资及原材料费用的柔性生产系统进行了研究。但这些算法主要解决的是产品纯加工或纯装配问题。对于实际生产中的多品种小批量的产品,特别是具有树状工艺结构的单件复杂产品(复杂单产品)的生产调度,如果分别进行产品的加工与装配,必然割裂产品生产过程中加工与装配内在的可并行处理的关系,影响产品制造效率。为此,文献[8]提出了同时考虑产品加工和装配一同处理的综合柔性调度的分步式算法,由于该算法采取最短加工时间为每个工序确定加工设备,有可能导致多个工序的加工设备相同,使负载较重的加工设备形成生产瓶颈。

文献[9]在文献[8]最短加工时间规则的基础上,通过将产品工艺树初始工序最早加工时间产生的静态重叠时间段上的工序调整至可柔性加工的设备上,增加了工序间的并行处理,同时减少某些设备上存在较多工序的情况。虽然文献[9]算法在一定程度上降低了设备的负载并减少瓶颈设备出现的可能,但由于产品的加工是一个动态调整的过程,加工前某些静态重叠时间段上的工序,在调度后未必重叠;某些加工前不属于静态重叠时间段上的工序在调度的过程中可能重叠。因此,将静态重叠时间段上的工序分配到不同设备上加工的并行处理效果有一定的局限性,影响产品的完成时间。

为了解决静态重叠时间段不能真实反映工序并行处理的问题,本文拟将实际调度中动态重叠的工序分配到不同的设备上并行处理,提高工序的实质并行处理率。由于空闲设备驱动^[10-11]策略可较好地设备驱动时刻动态发现可同时并行加工的工序,于是本文采用空闲设备驱动策略调度工序,当空闲设备存在可重叠加工的工序时,将可重叠加工

的工序动态地分配给其他可加工设备并行处理,实现实际重叠工序的并行处理并预防瓶颈设备出现。

由于柔性制造中工序加工设备的不唯一性,使得每个设备上分配到的工序不确定,为了设计使工序充分并行和缩短产品完成时间的复杂单产品综合柔性调度算法,本文对空闲设备驱动时刻可重叠加工的工序分别采用并行优化分配策略和最早加工结束策略进行调度分配:当某空闲设备上存在多个可计划加工工序且其中的某些可计划加工工序的其他可柔性加工设备也空闲,即这些可计划加工工序可以同时并行加工处理时,提出了并行优化分配策略,将这些可计划加工工序分配到不同的设备上并行加工处理,尽量保证各空闲设备上分配的工序的加工时间最小,对可计划加工工序不能并行加工处理只能串行加工的情况,为了合理地安排该空闲设备上工序的加工顺序,避免该设备上工序的串行加工顺序不当出现较长实质路径影响调度结果的情况,采用了动态实质短路径策略对工序选择调度^[12];当某空闲设备上不存在可计划加工工序但存在可调度加工的工序时,为了避免该空闲设备处于闲置状态,提出了最早加工结束策略,从可调度工序集中为该空闲设备寻找最早加工结束的工序。实例表明,本文提出的算法可以提高复杂单产品综合柔性调度问题的并行处理率,缩短产品的制造时间。

1 调度方案设计

为了方便加工和装配综合柔性调度的处理,一般将加工处理和装配处理统一为加工,加工设备和装配设备统一为设备。所研究的综合柔性调度问题满足约束条件:工序之间有次序约束关系,每道工序都必须在其紧前工序加工结束后或没有紧前工序时才可开始加工;每道工序可以在多个设备上加工,工序的加工时间随加工设备的不同而不同;

每台设备在某一时刻只能加工一道工序且加工过程不能中断;不存在相同设备。为了简化综合柔性调度问题的描述,相关概念的定义如下所述。

1.1 相关概念的定义

定义 1(计划加工设备):指每个工序的最短加工时间的柔性可加工设备。

定义 2(计划加工工序):计划加工设备上的工序。

定义 3(可调度计划加工工序):设备驱动时刻,空闲设备上可加工的计划加工工序。

定义 4(备选柔性工序集):指可在设备驱动时刻开始加工且加工结束时间比在计划加工设备上早

的计划加工工序组成的集合。

定义 5(并行分配工序集): 设备驱动时刻, 某空闲设备上的可以分配到其他空闲设备上并行加工处理的可调度计划加工工序组成的集合。

1.2 调度策略设计

研究加工和装配综合柔性调度算法, 就是为加工和装配可在多个设备上一同处理的工序选择合适的加工设备, 在满足工序间前后约束关系的前提下, 使产品的完工时间尽可能地短, 即产品尽早加工结束。根据前言分析, 本文设计的算法应考虑三个要求: ① 动态重叠加工工序分解; ② 串行加工工序合理排序; ③ 空闲设备忙。于是, 设计和采用了相应的三个调度策略: ① 并行优化分配策略; ② 实质短路径策略; ③ 最早加工结束策略。

1.2.1 并行优化分配策略

当某一空闲设备上存在多个可同时分配到其他空闲设备上加工处理的可调度计划加工工序时, 将这些工序分配到其他空闲设备上并行加工处理; 如果该空闲设备上最后仍存在一个以上的可调度计划加工工序时, 采用动态实质短路径策略, 确定该设备上的加工工序。通过使动态重叠加工工序并行处理和减少串行加工, 缩短产品完工时间。

1.2.2 实质短路径策略

由文献[12]知, 当多个可调度工序必须在同一设备上串行时, 优先加工某一可调度工序, 必然导致同设备上的其他可调度工序延后加工, 进而使其其他可调度工序所在的路径上的所有工序延后加工, 形成实质调度路径。由于可调度工序所在的路径长度不同, 因此会产生不同长度的真正影响产品加工时间的实质路径。实质路径计算方法是将路径最长的计划调度工序与可调度工序集中其他同设备工序两两交叉放到对方路径之前, 计算形成的实质路径长度。由于实质路径长度越长产品完成时间越长, 于是实质短路径的策略是优先调度实质短路径上的工序。

1.2.3 最早加工结束策略

对并行优化分配策略处理的计划加工工序之外的可调度工序, 考虑空闲设备尽量忙的要求, 设计了最早加工结束策略。当某空闲设备上不存在可调度计划加工工序时, 为避免该空闲设备闲置和减轻可调度工序集中的工序排队等待各自的计划加工设备的压力, 从并行优化分配策略处理的计划加工工序之外的可调度工序集中, 寻找能在此空闲设备上加工且加工结束时间比在原计划加工设备上的加工结束时间早的工序。实现计划加工工序和计划加工工序之外的可调度工序充分并行和尽早结束, 有

效地避免了空闲设备的闲置问题, 使设备得到均衡利用。

1.3 基于设备驱动和实质路径的动态并行综合柔性调度方案设计

首先按最短加工时间确定各设备上的计划加工工序; 再采用空闲设备驱动策略, 设备驱动时刻, 当某空闲设备上存在多个可调度计划加工工序且其中的某些可调度计划加工工序的其他可柔性加工设备也空闲时, 采用并行优化分配策略对工序进行分配调度; 当某空闲设备上不存在可调度计划加工工序但可调度工序集中存在可同时并行处理的可调度工序时, 采用最早加工结束策略选择调度工序。

1.4 基于设备驱动和实质路径的动态并行综合柔性调度问题的数学模型

给定一个产品, 假定 N 为该产品的工序数量, M 为加工设备数量, m 为工序的计划加工设备, T_k ($0 \leq k \leq n$) 为设备结束加工刚空闲的设备驱动时刻, t'_{id} 表示工序 i 在设备 D 上的加工时间, t_{im} 为第 i 道工序在计划加工设备 m 上的加工时间, T_{im} 为第 i 道工序在计划加工设备 m 上加工的设备驱动时刻, m^k 表示设备驱动时刻 T_k 时将要处理的空闲设备, E_i 为 T_k 时刻的备选柔性加工工序集, 当 m^k 上存在可调度计划加工工序 i 时(此时 m^k 与计划加工设备 m 是同一设备), $X_{im}^k=1$ (即 $X_{im}=1$), 否则 $X_{im}^k=0$; 当 m^k 上存在可调度计划加工工序以外的可调度工序时(此时 m^k 与计划加工设备 m 为不同设备), $X=1$, 否则 $X=0$ 。于是, 本文研究问题的数学描述为

$$T = \min \{ \max \{ T_k \} \} \quad (1)$$

$$\text{s.t. } t_{im} = \min \{ t'_{i1}, t'_{i2}, \dots, t'_{id} \} \quad (2)$$

$$t_{im} = (L_{jm} + t_{im}) < (L_{im} + t_{jm}) ? t_{im} : t_{jm} \\ X_{im} = 1 \quad (3)$$

$$E_i = (T_k + t_{im}^k) \leq (T_{im} + t_{im}) ? t_{im}^k : NULL \\ X_{im}^k = 0; X = 1; \quad (4)$$

$$t_{im} = \min \{ E_i \} \{ E_i \neq NULL \} \quad (5)$$

$$T_{nm} \geq T_{im} + t_{im} \quad (6)$$

$$T_{xy} \geq \max(T_{im} + t_{im}) \quad (7)$$

式(1)表示问题的目标函数 T 是产品最后完成工序的结束时间, 即产品的总加工时间 $\max(T_k)$ 尽可能短; 式(2)表示工序 i 的短用时计划加工设备为 t_{im} ; 式(3)表示采用动态实质短路径策略确定某空闲设备上的加工工序。其中 L_{im} 表示设备 m 上路径长度最长的计划调度工序 i 所在的路径长度, L_{jm} ($j=1$,

2, ..., n-1) 为与计划调度工序 i 同设备加工的可调度工序 j 所在的路径长度, 通过比较 $L_{im}+t_{jm}$ 和 $L_{jm}+t_{im}$ 的大小确定实质短路径, 将实质短路径上的可调度工序的加工时间赋值给 t_{im} 。式(4)表示搜索符合在空闲设备上的加工结束时间比在计划加工设备上的最早加工结束时间早的可调度工序, 并将其加入备选柔性工序集 E_i 中; 式(5)表示将 E_i 中加工结束时间最早的工序确定为该空闲设备上的加工工序, 并将其加工时间值赋值给 t_{im} ; 式(6)表示同一设备上, 后一道工序 n 的设备驱动时刻必须在当前工序 i 加工结束之后才能开始; 式(7)表示任意工序的开始加工时刻必须在其工艺约束的前序工序全部加工完毕后才能开始。

2 相关调度策略设计

2.1 并行优化分配策略

2.1.1 并行优化分配策略的设计思想

设备驱动时刻, 当某一空闲设备上存在可以分配到其他空闲设备上并行加工处理的可调度计划加工工序时, 先为该空闲设备上的可并行加工处理的可调度计划加工工序建立一个并行分配工序集; 将并行分配工序集中加工时间最短的工序预分配给该空闲设备; 再将并行分配工序集中的其他工序按加工时间由小到大分配到其他不存在可调度计划加工工序且加工时间少的空闲设备上并行处理。若处理后并行分配工序集中还有多个工序, 则采用动态实质短路径策略确定该空闲设备上的加工工序。

2.1.2 并行优化分配策略实现的具体步骤

(1) 在 T_k 设备驱动时刻, 计算存在多个可调度计划加工工序的空闲设备数 M_Δ 。

求 M_Δ 具体方法如下: T_k 时刻, 令 $M_\Delta=0$, 如果空闲设备在可调度工序集中有两个以上的计划加工工序, 则 $M_\Delta=M_\Delta+1$ 。

(2) 按设备号递增的顺序, 当设备 m_h 在 T_k 时刻存在 $N_\Delta(N_\Delta \geq 2)$ 个可调度计划加工工序 P_x ($N_\Delta \geq x$), 为 P_x 建立并行分配工序集 S_h , 即 $P_x \in S_h$ 。

求 N_Δ 具体方法如下: T_k 时刻, 令 $N_\Delta=0$, 比较可调度工序集中工序的计划加工设备属性, 当存在计划加工设备为空闲设备 m_h 时, $N_\Delta=N_\Delta+1$ 。

(3) 将 S_h 中的工序按在设备 m_h 上的加工时间由小到大的顺序排序, 若存在加工时间相同的工序, 则路径长的工序排在前面。

(4) 先将 S_h 中的第 1 个工序 P_1 预分配给 m_h ; 再将其他工序 P_x ($x \geq 2$) 分配到其他空闲设备上并行处理, 具体操作如下。

1) 对于工序 P_x ($x \geq 2$), 根据工序的加工设备属

性找出该工序的所有其他可柔性加工设备, 若未找到, 转步骤 5)。

2) 确定此驱动时刻可柔性加工设备是否空闲, 若均不空闲, 转步骤 5)。

3) 判断空闲的可柔性加工设备此时是否有可调度计划加工工序, 若均有可调度计划加工工序, 转步骤 5)。

4) 判断找到的可柔性加工设备是否唯一, 若唯一, 则将工序 P_x ($x \geq 2$) 在该可柔性加工设备上加工; 若不唯一, 比较工序 P_x ($x \geq 2$) 在所有找到的可柔性加工设备上的加工时间, 将工序 P_x ($x \geq 2$) 分配到其中加工时间最短的设备上加工。将工序 P_x 从可调度工序集中删除, 并将加工的空闲设备标记为非空闲。

5) $N_\Delta=N_\Delta-1$, 当 $N_\Delta=0$ 时, S_h 中的工序全部判断完毕, 转步骤(5), 否则转步骤 1)。

(5) 经过步骤(4)处理后, S_h 中除了 P_1 外还有其它工序, 则对 S_h 中剩余的工序采用实质短路径策略确定 m_h 在 T_k 时刻的加工工序; 否则 P_1 在 m_h 上加工。将确定的加工工序从可调度工序集中删除。

由文献[12]和实质短路径策略可知, 确定实质短路径的步骤如下: 首先将 P_1 与 S_h 中剩余工序两两交叉放到对方路径之前, 计算形成的实质路径长度, 调度实质短路径上的工序。

实质短路径策略实现方法如下: 设 P_j ($j=1, 2, \dots, k$) 为 S_h 中剩余工序, t_{1h} 和 t_{jh} 分别为工序 P_1 和 P_j 在设备 m_h 上的加工时间, L_{1h} 和 L_{jh} 分别为设备 m_h 上的工序 P_1 和 P_j 到根节点的路径长度, H_{1j} 和 H_{j1} 分别为工序 P_1 和 P_j 交叉相加所形成的实质路径长度, 即 $H_{1j}=L_{jh}+t_{1h}$, $H_{j1}=L_{1h}+t_{jh}$ 。选择长度最短的实质路径 $H_{xy}=\min\{H_{1j}, H_{j1}\}$ 为实质短路径, 调度实质短路径上的可调度工序; 若实质短路径存在多条, 则选择这些短路径上的可调度工序中加工时间最多的工序为实质调度工序。实质短路径策略流程图如图 1 所示。

(6) $M_\Delta=M_\Delta-1$, 当 $M_\Delta=0$ 时, 存在多个可调度计划加工工序的空闲设备处理完毕, 转步骤(7), 否则转步骤(2)。

(7) 结束。

2.1.3 并行优化分配策略实现的流程图

为了降低算法复杂度, 本文采取对所有设备进行 1 次循环判断存在可并行处理工序设备的方法, 相应的算法流程图如图 2 所示。

2.2 最早加工结束策略

2.2.1 最早加工结束策略设计思想

设备驱动时刻, 寻找能在某空闲设备上加工且结束时间比在其计划加工设备上早的工序, 并将找到的工序加入该空闲设备的备选柔性加工工序集,

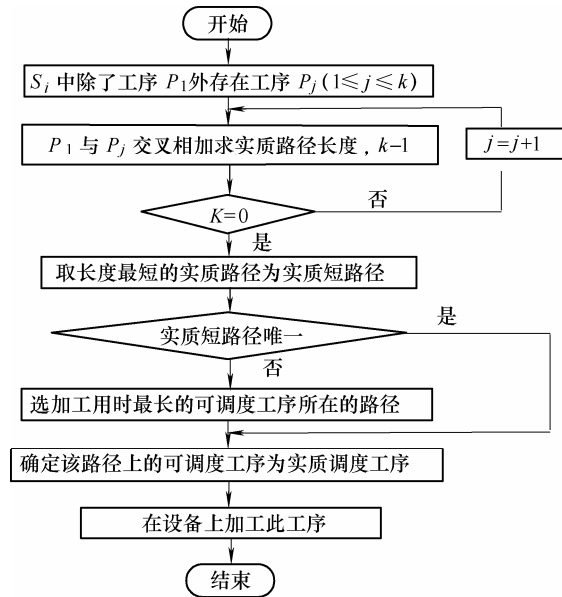


图1 实质短路径策略流程图

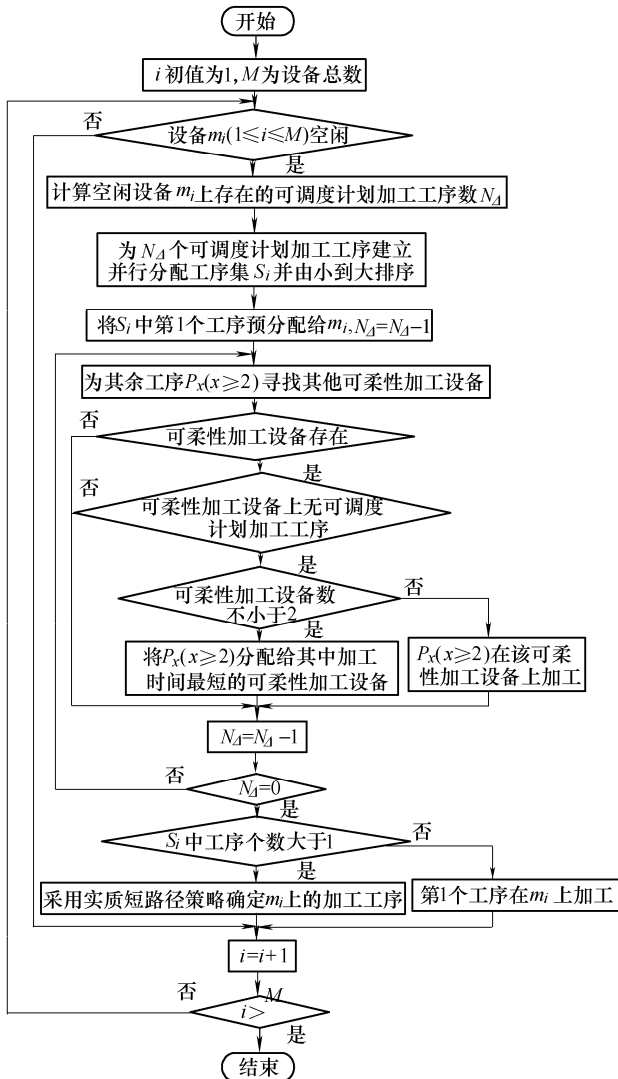


图2 并行优化分配策略流程图

再从备选柔性加工工序集中选择加工结束时间最早的工序为该空闲设备当前加工工序。

2.2.2 最早加工结束策略实现的具体步骤

(1) 在 T_k 设备驱动时刻,按设备号递增的顺序,寻找可调度计划加工工序不存在的空闲设备,如果找到,转步骤(2),否则转步骤(8)。

(2) 对找到的满足条件的设备 Ω ,遍历可调度工序集中剩余的可调度工序(设 T_k 时刻可调度工序集中剩余的工序数为 N_k)。

(3) 判断可调度工序集中工序 P_i 的可柔性加工设备中是否有 Ω 。如果有,转步骤(4),否则转步骤(6)。

(4) 计算并比较工序 P_i 在设备 Ω 上的加工结束时间 T 和其在计划加工设备 m 上的结束时间 T' 。

1) 设此可调度工序 P_i ($1 \leq i \leq N_k$) 在 Ω 上的加工时间为 $t_{i\Omega}$, 在 m 上的开始加工时间 T_{im} , 相应的加工时间 t_{im} , 则 $T=(T_k+t_{i\Omega})$, $T'=(T_{im}+t_{im})$ 。

2) 比较 T 和 T' 的大小,如果 $T' \leq T$,转步骤(6)。

(5) 将工序 P_i 加入设备 Ω 的备选柔性加工工序集 S_Ω 。

(6) $N_k=N_k-1$,若 $N_k=0$,此时可调度工序集中的工序遍历完毕,转步骤(7),否则转步骤(2)。

(7) 选择备选柔性加工工序集 S_Ω 中加工结束时间最早的工序在设备 Ω 上加工,如果存在最小加工结束时间值相同的情况,遵循长路径优先原则^[13]选择其中路径长度最大的作为加工工序并将该加工工序从可调度工序集中删除。

(8) 所有设备判断完毕,转步骤(9),否则转步骤(1)。

(9) 结束。

最早加工结束策略的流程图如图3所示。

3 算法设计

(1) 设置工序属性: 用 P_i 表示工序 i ; 用 T_i' 表示 P_i 的加工时间集; 用 M_i 表示 P_i 的加工设备集; 用 M_i' 表示 P_i 的计划加工设备; 用 C_i 表示 P_i 的紧前工序个数; 用 L_i 表示 P_i 到根节点的路径长度; 用 N_i 表示 P_i 的唯一紧后工序,当该值取0时表示 P_i 为根节点工序;于是,工序属性可以表示为 $P_i/T_i'/M_i/M_i'/C_i/L_i/N_i$ 。

(2) 设置设备状态属性,用 $busy$ 信号表示, $busy=0$,表示设备空闲, $busy=1$ 表示设备忙。

(3) 最短加工时间规则确定工序的计划加工设备,并计算紧前工序个数和路径长度。

1) 最短加工时间规则: 设工序 P_i 可在 m 个设备上加工,取最短加工时间 $\min\{t_{i1}, t_{i2}, \dots, t_{im}\}$ 的设备为工序 P_i 的计划加工设备。

2) 遍历所有工序的紧后工序属性 N_i 确定每个

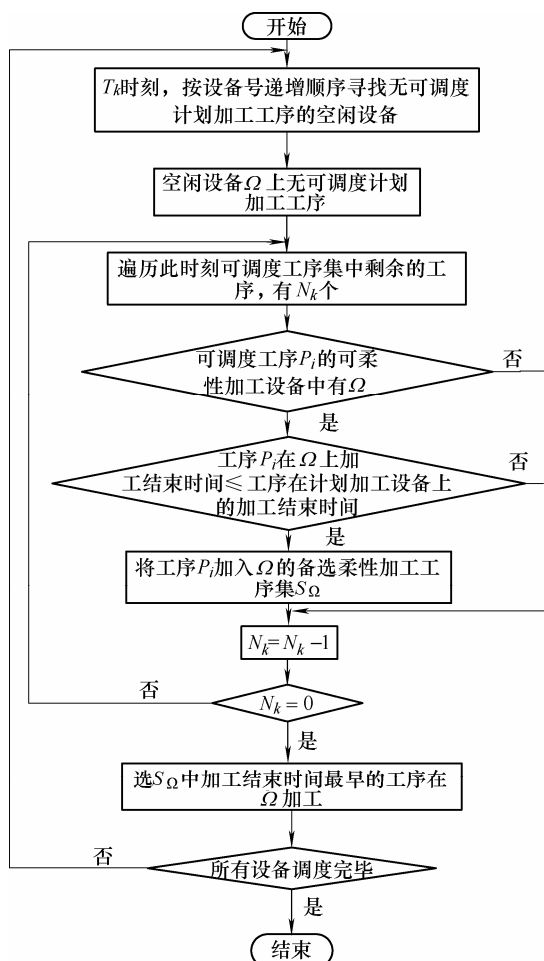


图 3 最早加工结束策略流程图

工序的紧前工序个数 C_i 。

初始时 $C_i=0$; $1 \leq i \leq n$

for $j=1$ to n {if $(N_j == P_i)$ $C_i = C_i + 1$; }。

3) 计算所有工序的路径长度 L_i 。根节点工序的路径长度为根节点工序的加工时间, 其余节点工序的路径长度 L_i 均可以由其紧后工序的路径长度 L_j 加上该节点工序的加工时间 T_i , 即 $L_i = L_j + T_i$ 得到。

(4) 空闲设备驱动时刻的调度。初始时刻 $T_0=0$ 。

1) 处理可调度工序集 S 。

初始时刻 T_0 , 由于叶节点工序无紧前工序, 所以叶节点工序属于 S 。

非初始时刻, 根据工序 P_i 的紧前工序个数属性 C_i , 将紧前工序个数 $C_i=0$ 的工序动态添加到可调度工序集 S 中。

2) 建立空闲设备集 M_{idle} 。

初始时刻 T_0 , 所有设备均处于空闲状态, 所以, 所有设备属于 M_{idle} 。

非初始时刻, 根据设备的状态属性, 将 $busy=0$ 的设备动态地添加到 M_{idle} 中。

3) 确定 T_k 时刻空闲设备上的加工工序。

步骤 1: 判断某空闲设备上是否存在可以并行

加工处理的可调度计划加工工序, 若是, 转步骤 2, 否则, 转步骤 4。

步骤 2: 采用并行优化分配策略, 具体算法过程见前面调度策略设计中的第 2.1.2 节, 将相应的加工设备的 $busy$ 信号置 1。

步骤 3: 按调度策略设计中的 2.1.2 采用并行优化分配策略处理后, 若某些空闲设备仍未分配到加工工序, 转步骤 4。否则转步骤 4)。

步骤 4: 采用最早加工结束策略。具体算法过程见前面调度策略设计中的第 2.2.2 节, 将相应的加工设备的 $busy$ 信号置 1。转步骤 4)。

4) 确定下一设备驱动时刻 T_k ($0 \leq k \leq n-1$)。

非初始时刻, 根据当前设备驱动时刻 T_k 的所有加工工序的加工结束时间, 将最早加工结束时间 $T_{k+1} = \min\{T_k + t_i\} (1 \leq i \leq m)$ 确定为下一设备驱动时刻。

将加工完毕的工序从可调度工序集中删除并将其紧后工序的紧前工序个数属性 C_i-1 。将加工完毕工序的加工设备的 $busy$ 信号标记为 0。

5) 判断可调度工序集中的工序是否全部加工完毕, 若是, 转步骤(5), 否则转步骤 1)。

(5) 结束。

算法流程图如图 4 所示。

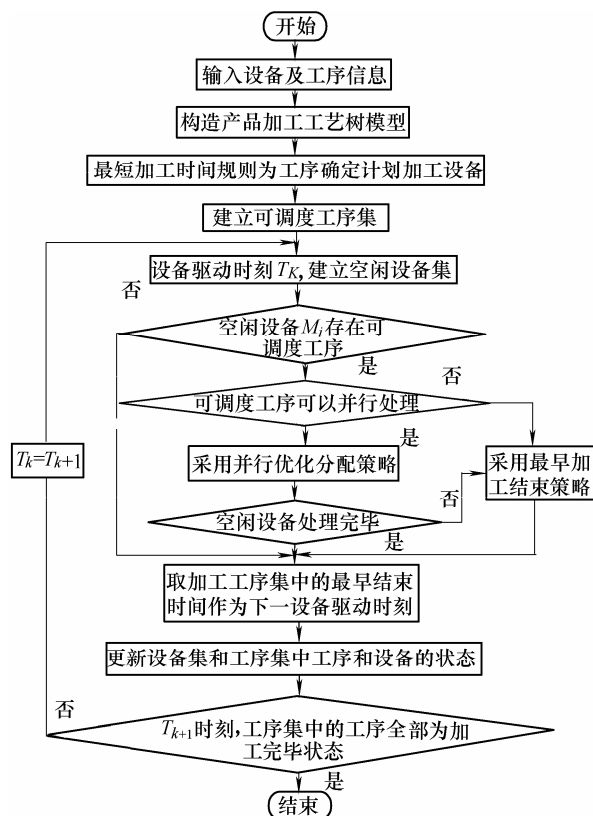


图 4 算法流程图

4 复杂度分析

设产品的工序总数为 n , 加工设备台数 m , 本

算法主要有如下操作。

(1) 最短加工时间规则确定计划加工设备。遍历 n 个工序的可柔性加工设备及对应的加工时间，由于每个工序最多可在 m 个设备上加工，最多需要比较 $m-1$ 次确定该工序的计划加工设备。因此，确定所有工序的计划加工设备最多需要 $n(m-1)$ 次操作。即时间复杂度为 $O(n)$ 。

(2) 计算所有工序的路径长度。根据文献[14]，确定所有工序路径长度的时间复杂度为 $O(n^2)$ 。

(3) 空闲设备驱动时刻的计算。选择加工工序集中的最小加工结束时间值作为下一次设备驱动时刻，最坏情况下，所有设备均处于忙碌状态，即加工工序集中有 m 道工序，需要比较 $m-1$ 次，由于最多有 n 个设备驱动时刻，所以最多需要比较 $n(m-1)$ 次，即时间复杂度为 $O(n)$ 。

(4) 并行优化分配策略确定加工工序如下所述。

1) 建立并行优化分配工序集：某一空闲设备 M_i 上最多存在 $n-1$ 个可并行处理的可调度计划加工工序，将其加入并行分配工序集 S_i 需 $n-1$ 次操作；对 S_i 中工序采用快速排序需要 $n \log_2 n$ 次操作。

2) 对 S_i 中的工序按序进行分配处理： S_i 中最多有 $n-1$ 个工序，将 S_i 中的第 1 个工序分配给该空闲设备 M_i ，需要 1 次操作；对于第 2 个工序，找出它的其它可柔性加工设备最多需要判断 $m-1$ 次，对每个其它可柔性加工设备判断是否有计划加工工序，最多需要判断 $m-1$ 次，最坏情况是第 2 个工序有 $m-1$ 个可柔性加工设备，此时判断可柔性加工设备存在计划加工工序的次数为 $(m-1)^2$ ；将该工序分配给这 $m-1$ 设备中加工时间最小的设备上需要比较 $m-2$ 次；由于此时最多有 $n-2$ 个可调度工序需要此操作，因此，对并行优化分配工序集中的工序进行分配处理操作次数最多为 $1+[(m-1)^2+(m-2)](n-2)$ 。

3) 动态实质短路径策略确定 M_i 上的加工工序。最坏情况下，所有空闲设备上均有可并行分配的工序，即对于每个空闲设备 M_i 而言，最多有 $n-1$ 个工序在 M_i 上串行加工，将第 1 个工序 P_1 分别与并行分配工序集里的工序 P_j 执行操作： $H_{jk}=L_k+t_j$ ，

$H_{kj}=L_j+t_k$ 需两次操作。由于并行分配工序集中除 P_1 外最多有 $n-2$ 个工序，最多需要执行加法操作 $2(n-2)$ 次，产生 $2(n-2)$ 条实质路径，从中选取实质路径值最小的需要比较 $2(n-2)-1$ 次。所以确定实质调度工序最多需要进行 $2(n-2)+(2(n-2)-1)=5n-11$ 次操作。由于最多有 m 个空闲设备，所以实质短路径策略最多需要操作 $m(5n-11)$ 次。

所以并行优化分配策略总共的操作步骤为 $n-1+n \log_2 n+(1+((m-1)^2+(m-2)))(n-2)m(5n-11)$ ，由于 $m \ll n$ ，所以复杂度为 $O(n^2)$ 。

(5) 最早加工结束策略确定加工工序。由于每个工序最多有 $m-1$ 个后补加工设备(除计划加工设备以外的其他可柔性加工设备)，因此，最多需要 $m-1$ 次比较操作确定某一可调度工序的后补加工设备中是否有此时的空闲设备；如果有，计算该工序在此空闲设备上加工结束时间 T 和在计划加工设备上的最早加工结束时间 T' ，需要 2 次操作；比较 T 和 T' 的大小需 1 次操作；如果 $T \leq T'$ ，将该工序其加入此时的可柔性加工工序集，需要 1 次操作；因此找到一个备选的加工工序需要 $m-1+2+1+1=m+3$ 次操作。最坏情况是该工序 $m-1$ 个后补加工设备都空闲，此时找到一个备选的加工工序需要 $(m-1)(m+3)$ 次操作。由于设备驱动时刻最多有 $n-1$ 个工序的后补加工设备中含有此时的空闲设备，所以找到所有的这些备选加工工序需要 $(n-1)(m-1)(m+3)$ 次操作。由于 $1 \leq m \ll n$ ，所以最早加工结束策略的时间复杂度为 $O(n)$ 。

综上所述，本文算法的复杂度为 $O(n^2)$ 。

5 实例对比与分析

以上提出的算法，没有考虑具体问题，因此，该算法具有普遍意义。为了帮助读者了解本文算法，下面通过一个实例说明。设有复杂单产品 A，该产品共有 18 个工序，分别在 4 台设备上加工。其加工工艺树如图 5 所示。图 5 中长方形框表示工序，其中的信息如下：工序名/加工设备名/加工时间。

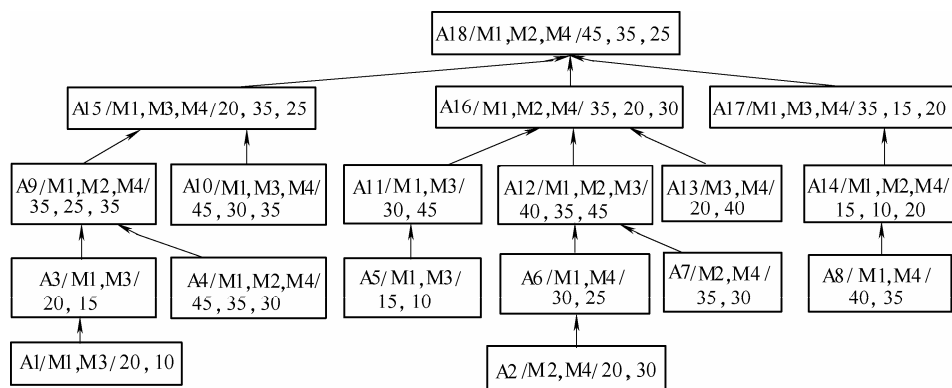


图 5 复杂单产品 A 的构造加工工艺图

下面分别采用基于静态加工时间的实质短路径法[9]和本文提出的算法对产品 A 进行调度。并对比调度甘特图说明算法特点。

5.1 基于静态加工时间的实质短路径法

按最短加工时间规则对产品 A 中的各工序确定计划加工设备,如图 6 所示。计算各设备上的有效空闲时间段、静态重叠时间段、静态重叠总时间和串行加工工序集,如表 1 所示。

根据每台设备的静态有效空闲时间段与其他设备静态重叠时间段的匹配情况确定工序的最终加

工设备。根据最终加工设备建立的工艺树如图 7 所示。

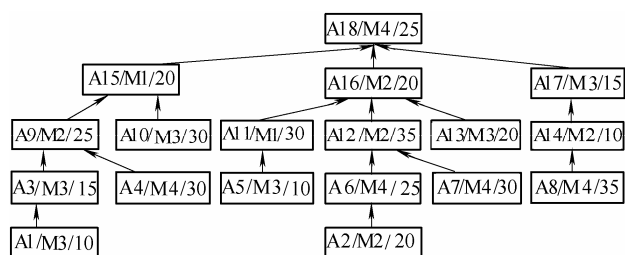


图 6 最短加工时间规则对复杂单产品 A 建立的加工工艺树

表 1 加工设备信息表

设备名	有效空闲时间段/h	静态重叠时间段/h	静态重叠总时间/h	串行加工工序集
M1	(0,10)(40,55)(75,100)	—	0	—
M2	(20,30)	(35,45)(45,55)	20	A9 A14 A12
M3	—	(0,10)(0,10)(0,20)(10,25)	55	A1A5A10A13
M4	—	(0,30)(0,30)(20,35)	75	A4 A6 A7 A8

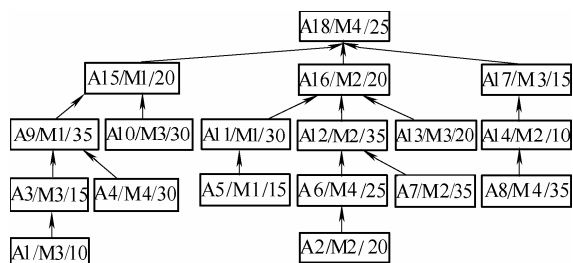


图 7 静态加工时间策略对产品 A 建立的加工工艺树

对图 7 的产品工艺树采用实质路径法调度,初始时备选工序集中工序 A2 所在的路径最长,确定 A2 为预调度工序,工序 A7 与 A2 为同设备加工的工序,实质路径长度为($H_{2,7}=135$, $H_{7,2}=155$),由于 A2 所在的实质路径长度 $H_{2,7}$ 最短,所以优先调度 A2,在备选工序集中删除工序 A2,将新生成的叶子节点工序 A6 加入备选工序集。在新产生的备选工序集中,重新按上述过程对工序进行调度。调度次序为 A2、A7、A4、A1、A6、A3、A5、A8、A9、A12、A10、A11、A13、A14、A15、A16、A17、A18。调度甘特图如图 8 所示,总加工时间 145 个工时。

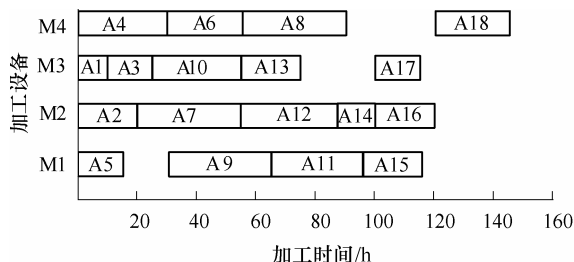


图 8 静态加工时间策略调度产品 A 的甘特图

5.2 基于设备驱动和实质路径的动态并行综合柔性调度算法

按最短加工时间规则对产品 A 中的各工序确定

计划加工设备,如图 6 所示。

初始时刻 $T_0=0$,可调度工序集中的工序为{ A1, A4, A10, A5, A2, A7, A13, A8 },空闲设备 M3 和 M4 存在多个可调度计划加工工序,为 M3 和 M4 建立并行分配工序集并排序: $S_3 \{ A1, A5, A13, A10 \}$, $S_4 \{ A7, A4, A8 \}$ 。

按序先处理空闲设备 M3 的并行分配工序集 S_3 ,将 S_3 中第 1 个工序 A1 分配给 M3;对于 S_3 中的第 2 个工序 A5, A5 的其他可柔性加工设备为 M1 且空闲,此时 M1 无可调度计划加工工序,将 A5 分配给 M1 上加工;对于第 3 个工序 A13, A13 的其他可柔性加工设备为 M4,由于 M4 此时有可调度计划加工工序,所以对 A13 不作处理, A13 仍在 S_3 中;对于第 4 个工序 A10, A10 的其他可柔性加工设备为 M1, M4,由于 M1 已经被分配工序, M4 有可调度计划加工工序,所以对 A10 不作处理, A10 仍在 S_3 中。于是 S_3 中剩余的工序为{ A1, A13, A10 },采用实质短路径法将 A1 分别与 A13, A10 交叉相加得实质路径长度: ($H_{1,13}=75$, $H_{13,1}=115$), ($H_{1,10}=85$, $H_{10,1}=125$), 两组数据中实质路径长度 $H_{1,13}=75$ 最短,所以确定 A1 为实质调度工序,调度 A1 到 M3 上加工。同理,按上述方法处理设备 M4 的工序,确定 A7 在 M4 上加工;由于空闲设备 M2 上只有一个可调度计划加工工序 A2,所以确定 A2 在 M2 上加工。

于是时刻 T_0 的各空闲设备处理完毕后,加工工序集中各工序的加工结束时间为{ A5 :15, A2 :20, A1 :10, A7 :30 },选择加工结束时间最早的时刻 $T_1=10$ 作为下一设备驱动时刻。

在 $T_1=10$ 时刻,设备 M3 空闲,可调度工序集

为 $\{A3, A4, A8, A10, A13\}$, 为设备 M3 建立并行分配工序集 $S_3 \{A3, A10, A13\}$, 将 A3 分配给 M3, 由于 A10 和 A13 的其他可柔性加工设备在此时可均未空闲, 所以 A10 和 A13 没有被分配出去仍在 S_3 中, 即 A3、A10 和 A13 只能在 M3 上串行加工, 将 A3 与 A10 和 A13 分别交叉相加, 得到实质路径长度为 $(H_{3,10}=90, H_{10,3}=105), (H_{3,13}=80, H_{13,3}=105)$, 两组数据中实质路径长度 $H_{3,13}=80$ 最短, 所以确定 A3 为 M3 上的加工工序, 计算加工结束时间为 $T=25$, 此时加工工序集中各工序的加工结束时间为 $\{A5:15, A2:20, A3:25, A7:30\}$, 选择加工结束时间最早的时刻 $T_2=15$ 作为下一设备驱动时刻。

在 $T_3=15$ 时刻, 设备 M1 空闲, 可调度工序集中的为 $\{A4, A8, A10, A11, A13\}$, M1 上的计划加工工序只有 A11, 所以确定 A11 为 M1 上的加工工序, 计算加工结束时间为 $T=45$, 此时加工工序集中各工序的加工结束时间为 $\{A11:45, A2:20, A3:25, A7:30\}$, 选择加工结束时间最早的时刻 $T_3=20$ 作为下一设备驱动时刻。

在 $T_2=20$ 时刻, 设备 M2 空闲, 可调度工序集为 $\{A4, A6, A8, A10, A13\}$, M2 不存在计划加工工序但存在可调度工序 A4, 采用最早加工结束策略, 计算 A4 在计划加工设备 M4 上的最早加工结束时间为 $30+30=60$, 在空闲设备 M2 上的加工结束时间为 $20+35=55$, 由于 $55<60$, 所以选择 A4 在设备 M2 上加工, 此时加工工序集中各工序的加工结束时间为 $\{A11:45, A4:55, A3:25, A7:30\}$, 选择加工结束时间最早的时刻 $T_4=25$ 作为下一设备驱动时刻。

按上述过程重新对可调度工序集中的工序进行调度。所确定的调度次序为: A2、A1、A7、A5、A3、A11、A4、A10、A6、A8、A12、A13、A9、A14、A16、A15、A17、A18。调度甘特图如图 9 所示, 总加工时间 140 个工时。

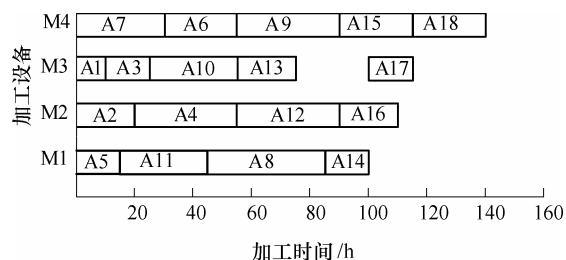


图 9 本文算法调度产品 A 的甘特图

通过对比甘特图可以看出, 在不提高算法复杂度的前提下, 采用本文算法对实例进行调度耗时为 140, 比算法 少了 5 个单位的加工时间; 因此

算法 优于算法 。

算法 之所以比算法 更优, 是因为算法 在设备驱动时刻, 采用并行优化分配策略, 将可调度计划加工工序多的空闲设备上的工序调度到无计划加工工序的空闲设备上, 分解了可调度计划加工工序多的空闲设备的负担, 实现工序并行处理。例如, 虽然算法 中工序 A8 在设备 M1 上的加工时间为 40, 比算法 中在短用时设备 M4 上的加工时间 35 长, 但由于算法 中工序 A8 在 M1 上的开始加工时间为 45, 早于算法 中工序 A8 在设备 M1 上的开始加工时间 55, 使得算法 中工序 A8 的加工结束时间 85 比算法 中工序 A8 的加工结束时间 90 早。即本文算法可使工序加工结束时间更早。

6 结论

本文针对复杂单产品综合柔性调度问题, 提出基于设备驱动和实质路径的复杂单产品动态并行综合柔性调度算法。该算法在不增加复杂度的情况下, 增加了工序间的并行处理, 提高了设备利用率, 缩短产品完工时间, 结论如下。

- (1) 采用空闲设备驱动策略, 方便动态寻找实质可重叠加工的工序。
- (2) 提出的并行优化分配策略可将重叠加工的工序并行加工处理。
- (3) 采用实质短路径策略缩短产品完成时间。
- (4) 提出的最早加工结束策略充分利用设备资源并增加工序并行处理。
- (5) 由于产品零件的工艺结构也是树状的, 所以多零件复杂产品的工艺结构也是树状的, 即本文提出的调度算法不限制产品包含零件的数量。

本文提出的算法对进一步研究多目标综合柔性调度问题有一定的借鉴作用。

参 考 文 献

- [1] 张超勇, 饶运清, 李培根, 等. 柔性作业车间调度问题的两级遗传算法[J]. 机械工程学报, 2007, 43(4): 119-124.
ZHANG Chaoyong, RAO Yunqing, LI Peigen, et al. Bilevel genetic algorithm for the flexible job-shop scheduling problem[J]. Chinese Journal of Mechanical Engineering, 2007, 43(4): 119-124.
- [2] XING Lining, CHEN Yingwu, YANG Kewei. Multi-objective flexible job shop scheduling: Design and evaluation by simulation modeling[J]. Applied Soft Computing, 2009, 9(1): 362-376.

- [3] XIE Shiman. A new genetic algorithms combined with learning strategy for flexible job-shop scheduling problem[J]. *Advances in Intelligent and Soft Computing*, 2012, 117: 153-161.
- [4] GHASEM M, MEHDI M. A pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search [J]. *International Journal of Production Economics*, 2011, 129(1): 14-22.
- [5] 施进发, 焦合军, 陈涛. 交货期惩罚下柔性车间调度多目标 Pareto 优化研究[J]. *机械工程学报*, 2012, 48(12): 184-192.
- SHI Jinfa, JIAO Hejun, CHEN Tao. Multi-objective pareto optimization on flexible job-shop scheduling problem about due punishment [J]. *Journal of Mechanical Engineering*, 2012, 48(12): 184-192.
- [6] 张国辉, 高亮, 李培根, 等. 改进遗传算法求解柔性作业车间调度问题[J]. *机械工程学报*, 2009, 45(7): 145-151.
- ZHANG Guohui, GAO Liang, LI Peigen, et al. Improved genetic algorithm for the flexible job-shop scheduling problem[J]. *Journal of Mechanical Engineering*, 2009, 45(7): 145-151.
- [7] 鞠全勇, 朱剑英. 多目标批量生产柔性作业车间优化调度[J]. *机械工程学报*, 2007, 43(8): 148-154.
- JU Quanyong, ZHU Jianying. Multi-objective flexible job shop scheduling of batch production[J]. *Chinese Journal of Mechanical Engineering*, 2007, 43(8): 148-154.
- [8] XIE Zhiqiang, HAO Shuzhen, YE Guangjie, et al. A new algorithm for complex product flexible scheduling with constraint between jobs[J]. *Computers & Industrial Engineering*, 2009, 57(3): 766-772.
- [9] XIE Zhiqiang, WANG Peng, CHANG Ningning, et al. Flexible integrated scheduling algorithm base on static processing time[C]//2011 International Conference on Computer Science and Service System, CSSS 2011, 2011: 1279-1282.
- [10] 谢志强, 辛宇, 杨静. 基于设备空闲事件驱动的综合调度算法[J]. *机械工程学报*, 2011, 47(11): 139-147.
- XIE Zhiqiang, XIN Yu, YANG Jing. Integrated scheduling algorithm based on event driven by machines' idle[J]. *Journal of Mechanical Engineering*, 2011, 47(11): 139-147.
- [11] 谢志强, 辛宇, 杨静. 可回退抢占的设备驱动综合调度算法[J]. *自动化学报*, 2011, 37(11): 1332-1343.
- XIE Zhiqiang, XIN Yu, YANG Jing. Machine-driven integrated scheduling algorithm with rollback-preemptive [J]. *Automatica Sinica*, 2011, 37(11): 1332-1343.
- [12] XIE Zhiqiang, WANG Peng, GUI Zhongyan, et al. Integrated scheduling algorithm based on dynamic essential short path[J]. *Advances in Intelligent and Soft Computing*, 2012, 169: 709-715.
- [13] 谢志强, 张磊, 杨静. 基于调度长路径的复杂产品综合调度算法[J]. *计算机科学*, 2010, 37(2): 150-153.
- XIE Zhiqiang, ZHANG Lei, YANG Jing. Integrated scheduling algorithm of complex product based on scheduling long-path[J]. *Computer Science*, 2010, 37(2): 150-153.
- [14] 谢志强, 杨静, 杨光, 等. 可动态生成具有优先级工序集的动态 Job-shop 调度算法[J]. *计算机学报*, 2008, 31(3): 502-508.
- XIE Zhiqiang, YANG Jing, YANG Guang, et al. Dynamic job-shop scheduling algorithm with dynamic set of operation having priority[J]. *Chinese Journal of Computers*, 2008, 31(3): 502-508.

作者简介: 谢志强(通信作者), 男, 1962 年出生, 博士后, 教授, 博士研究生导师, CMES 和 CCF 高级会员。主要研究方向为企业智能计算与调度优化、数据库与知识工程。

E-mail: xiezhiqiang@hrbust.edu.cn