

# 基于 LSF 算法与动态关键路径的工序级任务调度问题的研究<sup>\*</sup>

王池平,时 轮,王 鹤  
(上海交通大学 机械与动力工程学院,上海 200240)

**摘要:**针对目前某研究所工序的实际生产时间较调度时间滞后,设备利用率不高导致的任务完工时间较长的情况,提出了 LSF(Least Slack First,空闲时间最短最优先)算法与动态关键路径法相结合的工序级任务调度算法。在工序排程的过程中应用 LSF 算法可以使关键路径任务之间的空闲时间尽可能的减小,再结合动态关键路径法则可以使生产周期尽可能压缩。该研究所的实际应用表明,上述针对工序级任务的调度算法可以有效的提高设备利用率和解决实际生产同调度之间的滞后问题,从而可以最小化完工时间,提高生产效率。  
**关键词:**LSF 算法;动态关键路径;工序级任务;最小化完工时间  
**中图分类号:**TH162;TG506      **文献标识码:**A

**Research on Process Scheduling Problem Based on LSF Algorithm and Dynamic Critical Path**  
WANG Chi-ping, SHI Lun, WANG He  
(School of Mechanical Engineering, Shanghai JiaoTong University, Shanghai 200240, China)

**Abstract:** In order to solve the problem that the actual production time of a research institute is longer than that of the scheduling time and the equipment utilization rate is not high, the LSF algorithm and the dynamic critical path method are proposed. Application of the LSF algorithm in the process of scheduling can make the idle time among tasks in critical path as short as possible, combined with the dynamic key path rule can make the production cycle shorter. The practical application of the institute shows that the above scheduling algorithm for process-level tasks can effectively improve the utilization of equipment and solve the problem of lag between actual production and scheduling, which can minimize the completion time and improve the production efficiency.  
**Key words:** LSF algorithm; dynamic critical path; process-level task; minimize the completion time

## 0 引言

调度是生产过程中的核心组成部分,其用于解决生产中怎么做的问题。它需给出各道工序何时、何地由何人作业,以达到指定性能指标的最优。调度问题中完工时间最短是优化指标中的重点之一,也是生产中提高生产效率最关键的性能指标之一。此类问题优化建模往往是非线性与多约束的,是 NP-hard 问题之一。

基于此,比较传统的求解算法有启发式遗传算法和设置较多限制的精益化算法。近年来,一些学者提出了相应的进化算法,利用前插或后插手手段调整设备上的工序顺序,最终完成优化目标,但没有从纵向考虑工序路径上的优化。此后,相关研究人员在研究前人对工序级任务调度算法的基础上,提出了始终基于动态关键路径法的调度算法。即先将工序集分配到各个

设备,然后在满足工序之间的约束条件的情况下,每次选取路径最长的任务优先调度,却没有横向考虑各设备上无前后约束的工序不合理排序带来的空闲时间而造成完工时间延长的问题。近两年(如文献[1-2])则主要是从设备维护和传统算法改进的角度考虑调度问题。

故而,本文结合最小化空闲时间的进化算法<sup>[3]</sup>(LSF 算法)和动态关键路径法<sup>[4]</sup>的优点,即不但考虑长路径带来更多任务并行的特点,也利用最小化空闲时间法对设备上工序进行顺序优化,最小化空闲时间,对工序级任务调度问题进行优化求解。

### 1 工序级任务调度问题描述

对于给定的任务集合  $S$ ,对  $S$  的一个调度是指  $S$  在时间轴上的一个排列,对于具体的问题需要考虑具

收稿日期:2017-07-30;修回日期:2017-08-31  
<sup>\*</sup>基金项目:上海市科学技术委员会项目(15111103403);国家自然科学基金项目(61203322)  
作者简介:王池平(1991—),男,安徽安庆人,上海交通大学硕士研究生,研究方向为智能调度与高级排产,(E-mail)18086486347@163.com。

体约束,综合起来就叫任务调度<sup>[5]</sup>。

本文对工序级任务调度综合了两个优化目标。它们是:①关键路径上各工序总加工时间最短;②各个设备在整个加工时段中空闲时间总和最少。最终目的即最小化完工时间。

关键路径上的空闲时间即指最长加工路径上相邻任务之间前面任务的结束时间和后面任务的开始时间之间有空隙。其原因是由于设备上的任务仅由动态关键路径法产生排列不合理造成的。动态关键路径法是指每次都选取路径最长的任务先加工的方法。

在该调度问题中,要满足以下几点<sup>[6]</sup>:

(1) 每个设备被当成不同的实体。

(2) 在同一时间,单个设备上只能有一个任务在进行。

(3) 每道工序必须在其所有关联的紧前工序<sup>[7]</sup>加工完成之后才能开始加工。

针对该调度问题建立数学模型如下:

优化目标:

$$\min( IT + \sum_{j=1}^n T_{pj} + \sum_{i=1}^k T_{Mi} ) \quad (1)$$

式中,  $IT$  表示总任务给定的起始时间,  $T_{pj}$  表示关键路径  $L$  上的第  $j$  道工序总加工时间(包括其准备时间),  $n$  表示该路径上共有  $n$  道工序,  $T_{Mi}$  表示关键路径  $L$  上机器  $i$  上的空闲时间,  $k$  表示共有  $k$  台设备。

约束条件:

$$TS_{i+1} \geq TS_i + W_i \quad (2)$$

(这类约束表示工序  $i+1$  的开始时间大于等于其前道工序  $i$  的开始时间与加工时间之和)

$$TS_n \geq \max( TS_i + W_i ) \quad (3)$$

(这类约束表示工序  $n$  的开始时间大于等于其前面所有工序中开始时间与加工时间之和的最大值)

$$C(m_i, m_k) * (TE_i - TS_k) * (TE_k - TS_i) \leq 0 \quad (4)$$

(这类约束表示同一时间同一设备上只能操作一个任务)

变量及符号说明:

$TS_i$  表示第  $i$  道工序开始时间。

$TE_i$  表示第  $i$  道工序结束时间。

$W_i$  表示第  $i$  道工序加工时间。

$C(m_i, m_k) = \begin{cases} 0 & \text{当 } m_i = m_k \text{ 时值为 1, 反之为 0,} \\ 1 & \end{cases}$   
 $m_k$  表示设备  $k$ 。

## 2 问题描述

结合工序之间的序偶关系以及产品的层次性,产品的加工工艺可以采用树状图描述,即加工工艺树。单个工件构成一个最简单的加工工艺树。本文所述的问题采用 4 台设备,4 个工件来研究优化过程,现实生产中可依此为模板研究有若干设备和若干工件的问题。首先,进行模型的建立,4 个工件分别构成一颗加工工艺树。其次,虚拟出一个根节点,让每颗工艺树成为它的孩子树,从而构成一颗大的加工工艺树(如图 1 所示)。加工工艺树的每个节点表示一个工序级的任

务,每个任务包含任务号,设备号以及加工时间。从根节点(没有上层任务的节点)到叶子节点(没有下层任务的节点)会构成一条条路径(其中路径最长的为关键路径)。某个节点处的路径长度即为根节点到此处该路径上所有任务加工时间之和。最终的关键路径由路径长度最大的节点所在的加工路线确定。

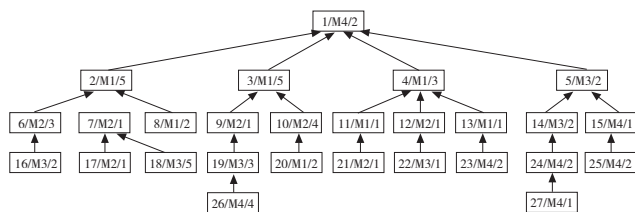


图 1 加工工艺树

基于以上理论,目前解决工序级任务调度问题的方法主要有拟关键路径法、动态关键路径法以及一些进化算法。其中动态关键路径法虽然在横向和纵向都对工序的优先级进行了考虑,并可以使得不同设备上并行任务尽可能的多,缩短总加工时间,但该算法会使得一旦关键路径上的任务之间空闲时间较大,就会对总加工时间产生较大的影响。另一方面,一些进化算法虽然可以使任务调度时关键路径上的空闲时间达到最优,但没有考虑任务之间的约束关系和层次关系。因此,本文采用一种进化算法(LSF 算法)和动态关键路径法相结合的方式解决工序级任务调度问题,以期能够最小化完工时间<sup>[8]</sup>。

## 3 算法设计

本研究所使用的 LSF 算法为进化算法中的一种,力求在基于该算法的思想基础上采用更简化的方法最小化空闲时间。例如,以路径为线索,先考虑关键路径,再考虑非关键路径,并且只采用前插手段进行优化<sup>[9]</sup>。后续的研究可以在动态关键路径法的基础上,采用前插和后插相结合的 LSF 算法,以期进行进一步优化。当然,鉴于本文主要考虑的只是工序级任务调度的最小化完工的优化问题,没有考虑设备上的产能均衡问题。实际生产中,任务调度是要综合考虑最小化完工问题和产能均衡问题的。产能均衡问题可以结合关键路径法和任务层次网络法来优化解决,即可以通过在中长期计划过程中做好任务规划并结合调度过程中的关键路径法来解决,本文在此不做详细介绍,具体可参考文献[5]。

针对工序级任务调度问题设计算法具体步骤如下:

步骤①:输入工件信息<sup>[10]</sup>(工序集  $PS$ )和设备信息(设备集  $MS$ );

步骤②:根据 BOM 表、加工工艺及设备信息为每个工件建立加工工艺树  $T_i$ ;

步骤③:将各个工件的加工工艺树整合成一颗总的加工工艺树  $T$ ,根结点记为  $Root$ ,各个工件的根结点  $pRoot$  为其孩子节点;

步骤④:按路径长度为每个任务分配优先级,并在原有的任务节点( $ID, WID, WT$ )上附上路径长度  $PL$ ,

记为 (ID, WID, WT, PL), 若任务之间的路径长度一样, 则加工时间短的任务优先级更高, 若加工时间又一样, 则优先级一样。

步骤⑤: 将任务分配到对应的设备上, 建立每个设备的任务集 (MSet)。

步骤⑥: 每次选取优先级最大的任务先加工, 直至每个设备上的任务集均为空。最后记录最大路径长度  $PL_{\max}$ 。

步骤⑦: 根据任务之间的关系, 形成任务关系矩阵  $G$ , 如任务 1、2、3、4、5, 形成如下矩阵<sup>[11]</sup> (0 表示任务与其自身的关系,  $\infty$  表示任务之间无承接关系, 具体数值表示该道任务到其后继任务的路径长度):

表 1 任务关系矩阵(示例)

任务 \ 后继任务					
	1	2	3	4	5
1	0	$\infty$	$\infty$	$\infty$	$\infty$
2	5	0	$\infty$	$\infty$	$\infty$
3	5	$\infty$	0	$\infty$	$\infty$
4	3	$\infty$	$\infty$	0	$\infty$
5	2	$\infty$	$\infty$	$\infty$	0

步骤⑧: 结合最大路径长度  $PL_{\max}$  和任务关系矩阵  $G$ , 依次搜索出关键路径上任务, 如本例中任务 26 处为最大路径长度  $PL_{\max}$ , 该任务为首个关键任务, 再结合关系矩阵依次搜索出任务 9、3、1, 再逆序排好记为集合  $GTask$ 。

步骤⑨: 确定搜索区间: 根据关键任务的分配开始生产时间  $st$ , 即计划排定的开始生产时间, 关键任务的最晚开始生产时间  $lt$ , 即为其后继任务分配开始生产时间—该任务工作时间, 关键任务的最早结束时间  $et$ , 即其前驱任务的分配结束生产时间, 结合随机数指定搜索方向和搜索区间 ( $ros$ ), 指定方式由式 (5) 给出:

$$ros = \begin{cases} [et, st] & \text{if } rand(0,1) < (st - et)/(lt - et) \\ [st, lt] & \text{if } rand(0,1) \geq (st - et)/(lt - et) \end{cases} \quad (5)$$

步骤⑩: 做前插操作: 依次遍历集合  $GTask$ , 若搜索区间有任务  $p_1, p_2, \dots, p_n$ , 对当前任务依次前插到  $pn, \dots, p_2, p_1$  之前, 合法并且可以缩短完工时间则执行, 反之不执行。

步骤⑥~步骤⑩整个过程的目标表达式如下:

$$\text{Minimize: } Root\_end\_time = IT +$$

$$\sum_{i=0}^k (Task\_end\_time - Task\_satrt\_time) \quad (6)$$

说明: 上述表达式为目标函数, 即最终根节点任务的结束时间最早。IT 为整个加工工艺树的起始时间。后面为关键路径任务及跟其加工时间相邻任务的加工周期之和。

4 实验结果分析

以图 2 所示的加工工艺树为模型树, 代表一个由 4 个工件组合而成的一个总的工件。图 3 中的实验结果是仅采用动态关键路径法而得来的, 每次取出路径最长的叶子节点先加工, 路径长度一样时则采取短时策略<sup>[12]</sup>。显而易见, 图 3 为动态关键路径法的结果, 任务并行无有数据的效果, 但例如任务 2、11、13 排列

的不合理就迟滞了关键路径上任务的开始时间, 最终总的加工时间为 29 h。图 4 只采用 LSF 思想, 则可能会使非关键路径上的任务占主导地位还会导致任务之间的排列不合理, 如任务 7 就不该排在任务 18 前面, 任务 16 到根节点的路径和任务 22 到根节点的路径上任务的排列就严重干扰了关键路径上任务的排列<sup>[13]</sup>, 导致完工时间反而延长至 30 h。

如图 5 所示, 容易看出, LSF 思想与动态关键路径法的结合, 既可以充分利用短时策略, 还可以使关键路径上的任务尽可能提前, 并很自然地将工序和设备之间的约束关系考虑进去, 同时也使关键路径上的任务排列最优, 最终达到最小化空闲时间的目标。从时间轴上来看, 通过 LSF 算法和动态关键路径法相结合, 则可使上述加工工艺树所描述的工序级任务调度所需总加工时间为 23 h。

很明显, 后者不仅从加工工艺树的纵向和横向考虑调度问题, 还进行了最小化空闲时间处理, 与前两者相比较, 优化效果明显。

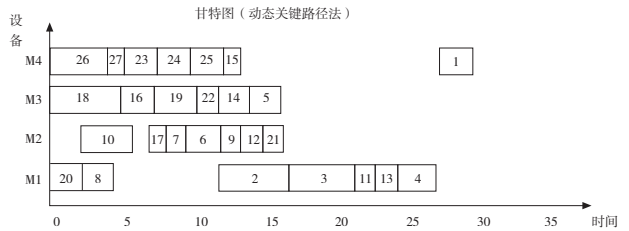


图 2 任务甘特图 1

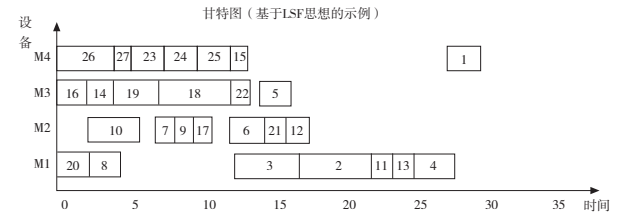


图 3 任务甘特图 2

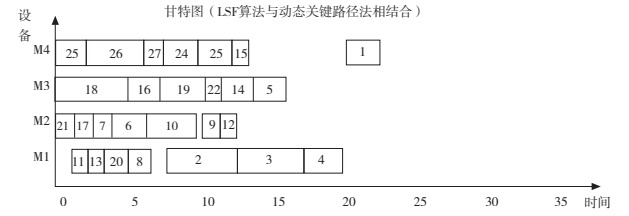


图 4 任务甘特图 3

5 结论

本文针对工序级任务调度问题提出了基于 LSF 算法和动态关键路径的调度方法, 以最小化完工时间为目标。采用某航天研究所的 4 个工件的加工工艺以及相应的 4 台设备进行效果评估。首先应用关键路径法得到初步的调度结果, 再使用 LSF 算法进行最小化空闲时间, 最后比较结果, 完工时间达到了更优化的效果。

[参考文献]

[1] 查靓, 金花, 潘志成, 等. 基于顺序相关调整时间的 FJSP 与设备维护计划集成优化[J]. 组合机床与自动化加工技术, 2016(5): 158 - 154. (下转第 184 页)



仅考虑最大完工时间最小时,图 8 和图 9 的两种调度方案均为该实例的最优解。考虑到现实生产中机器生产时碳排放不同,根据以上模型设置以下参数:

机器运行时的碳排放函数:

$$Y_{i1}, i = 1, 2, 3, 4, 5, 6 \quad (5)$$

机器空转时的碳排放函数:

$$Y_{i2}, i = 1, 2, 3, 4, 5, 6 \quad (6)$$

机器加工时的和空闲时的碳排放函数关系:

$$Y_{i1} = 3 \cdot Y_{i2}, i = 1, 2, 3, 4, 5, 6 \quad (7)$$

不同机器之间的碳排放函数关系:

$$Y_{1j} = \frac{1}{2} \cdot Y_{2j} = Y_{3j} = \frac{1}{3} Y_{3j} = 2 \cdot Y_{4j} = \frac{1}{2} \cdot Y_{5j} = \frac{2}{3} Y_{6j}, j = 1, 2 \quad (8)$$

结合以上公式,将图 8、图 9 中数据带入总碳排放函数公式中得到该实例的 2 种可行调度方案用机器  $M_1$  空闲时的碳排放函数表示,则得到总碳排放函数  $Y_{a总} = 255 \cdot Y_{12}$ ,  $Y_{b总} = 274 \cdot Y_{12}$ ,说明了图 8 的调度方案碳排放较小,在最大完工时间最小的情况下更加节能,方案 1 更优。

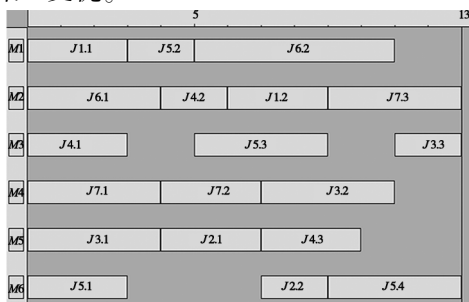


图 9 7 × 6 实例可行调度(方案 2)的甘特图

## 4 结论

(1)随着节能减排的需求不断增大,建立低碳排放的柔性作业车间可以减少企业生产成本,增加企业的竞争力。在考虑了最大完工时间最小的基础上以低碳排放为优化目标的作业车间调度问题,使问题更加接近生产的实际情况。

(2)通过建立模型将抽象的碳排放简化为可以求

解的具体问题,通过使用改进的遗传算法分析实例和研究企业的真实案例得出相应结论,并在不同的调度方案中分析了低碳排放的约束,说明了考虑低碳排放时的柔性作业车间调度问题的实用性,体现了设计的低碳模型和算法有效性。

(3)本文在建立的碳排放模型未充分考虑不同工件在不同机器上加工时的碳排放函数是不同的,且不同机器的碳排放函数为非线性关系的情况,下一步会考虑这些情况,使问题更加贴近车间的实际情况。

### [参考文献]

- [1] Zhang G, Gao L, Shi Y. An effective genetic algorithm for the flexible job-shop scheduling problem[J]. Expert Systems with Applications, 2011, 38(4):3563-3573.
- [2] 刘焱,刘长安. 基于混合遗传算法的车间动态调度研究[J]. 组合机床与自动化加工技术, 2017(3):158-160.
- [3] 刘韵,胡毅,罗企,等. 一种解决柔性车间作业调度问题的粒子群优化算法[J]. 组合机床与自动化加工技术, 2015,(12):144-147.
- [4] 徐华,张庭. 混合离散蝙蝠算法求解多目标柔性作业车间调度[J]. 机械工程学, 2016, 52(18):201-212.
- [5] 赵诗奎,方水良,顾新建. 柔性车间调度的新型初始机制遗传算法[J]. 浙江大学学报(工学版), 2013, 47(6):1022-1030.
- [6] 王超,黄波,贾春兰,等. 考虑加工质量和生产成本的柔性作业车间调度问题研究[J]. 机械, 2012, 39(12):10-14, 21.
- [7] 陈鸿海,赵韩. 实时工况驱动的柔性作业车间动态调度系统研究[J]. 合肥工业大学学报(自然科学版), 2016, 39(11):1441-1445,1455.
- [8] Wang Y M, Yin H L, Qin K D. A novel genetic algorithm for flexible job shop scheduling problems with machine disruptions[J]. International Journal of Advanced Manufacturing Technology, 2013, 68(5-8):1317-1326.
- [9] 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, 2009, 45(7):145-151.
- [10] 金亮亮,张超勇,邵新宇. 基于文化基因算法的集成工艺规划与调度研究[J]. 华中科技大学学报(自然科学版), 2017, 45(3):1-7. (编辑 李秀敏)

(上接第 179 页)

- [2] 李运霞,杜娟,孙王路. 基于多种群遗传算法的路径柔性车间调度问题[J]. 组合机床与自动化加工技术, 2014(3):152-155.
- [3] 张晶,孙少杰,范洪博,等. 一种高回报的最小空闲时间优先实时调度改进算法[J]. 计算机工程, 2017, 43(3):57-62.
- [4] Qiyi Li, Lei Wang, Lun Shi, et al. A Data Based Production Planning Method for Multi-variety and Small-batch Production[J]. International Conference on Big Data Analysis, 2017, 2(1):420-425.
- [5] 余建军,张定超,周铭新. 生产调度研究综述[J]. 机械设计与制造工程, 2009, 38(17):13-17.
- [6] Deming Lei, Xiuping Guo. Variable neighbourhood search for dual-resource constrained flexible job shop scheduling[J]. International Journal of Production Research, 2014, 52(9):2519-2529.

- [7] 谢志强,杨静,周勇,等. 基于工序集的动态关键路径多产品制造调度算法[J]. 计算机学报, 2011, 34(2):408-410.
- [8] 苏子林. 车间调度问题及其进化算法问题分析[J]. 机械工程学, 2008, 44(8):244-246.
- [9] 苏子林. 最小化空闲时间的 Job-shop 调度进化算法[J]. 计算机工程与应用, 2007, 43(27):79-81.
- [10] 常宁宁. 基于设备工序紧凑的综合调度算法研究[D]. 哈尔滨:哈尔滨理工大学, 2011.
- [11] 刘民. 基于数据的生产过程调度方法研究综述[J]. 自动化学报, 2009, 35(6):785-806.
- [12] 苗文明,陈泳,陈关龙,等. 面向延迟制造的 MRP 动态调整方法研究[J]. 自动化学报, 2008, 34(8):950-956.
- [13] 马雪丽,黄学文,曹德弼. 集成化车间作业计划动态关键路径编制算法[J]. 计算机集成制造系统, 2013, 19(9):2178-2186. (编辑 李秀敏)