

文章编号: 1001-0920(2011)07-1109-04

基于DBR理论的柔性流水车间动态调度

杨琴^{1,2}, 周国华¹, 林晶晶¹, 赵茜²

(1. 西南交通大学 经济管理学院, 成都 610031; 2. 四川师范大学 商学院, 成都 610101)

摘要: 针对柔性流水车间动态调度求解困难的问题, 首先分析调度问题的特征, 构建问题模型; 然后运用DBR(鼓-缓冲器-绳子)理论对问题进行分解简化, 并采用混合重调度策略和启发式算法进行动态调度; 最后建立瓶颈和非瓶颈资源调度的协调机制, 实现问题求解. 仿真实例表明, 所提出的算法是可行而有效的.

关键词: DBR理论; 柔性流水车间; 动态调度

中图分类号: F406.2

文献标识码: A

Dynamic scheduling of flexible flow shop problem based on DRB theory

YANG Qin^{1,2}, ZHOU Guo-hua¹, LIN Jing-jing¹, ZHAO Xi²

(1. School of Economics and Management, Southwest Jiaotong University, Chengdu 610031, China; 2. Commercial College, Sichuan Normal University, Chengdu 610101, China. Correspondent: YANG Qin, E-mail: yyuu2006@yahoo.com.cn)

Abstract: For the difficulties of flexible flow shop(FFS) dynamic scheduling problem, the problem's character is analyzed, and the model of the problem is built. Then based on the drum-buffer-rope(DBR) theory, the problem is decomposed and simplified. The dynamic scheduling is done by composite rescheduling policy and heuristic algorithm. Finally the coordinating rules of scheduling plan between bottleneck and non-bottleneck resource are built to solve the problem. An example shows the feasibility and effectiveness of the algorithm.

Key words: drum-buffer-rope theory; flexible flow shop; dynamic scheduling

1 引言

柔性流水车间(FFS)是流水车间和并行机环境的一般化形式, 取代多个串行设备的是多个串行工位, 每个工位包含许多并行设备, 许多学者运用启发式算法研究该问题^[1-2], 但大多针对静态问题. 而动态FFS问题更能体现实际生产的特征, Pinedo等人^[3-5]指出, 解决动态调度问题的有效方式是结合突发事件对原计划进行重新调整, 即再调度. 但是, 再调度将一个FFS问题变成多个, 增加了求解的难度.

DBR理论是基于约束理论的生产作业与控制方法, DBR是drum-buffer-rope(鼓-缓冲器-绳子)的简称, 该理论强调通过对瓶颈的优化, 简化大规模复杂调度问题^[6]. 目前, 已有学者研究如何运用基于DBR理论的调度算法对静态调度问题进行求解^[7-10], 并取得了较好的结果, 但涉及动态和柔性调度问题的研究文献还比较少. 本文试图在已有研究的基础上, 通过瓶颈

和非瓶颈问题的分解与协调来简化问题, 结合重调度策略和启发式算法实现动态问题求解, 并通过设置缓冲和缓冲管理来协调瓶颈与非瓶颈的关系, 确保调度可行.

2 问题描述及建模

下标 j 表示任务($j=1, 2, \dots, n$), n 表示任务数量; 上标 i 表示工位($i=1, 2, \dots, l$), l 表示工位数量; 上标 h 表示工位中的并行异速机($h=1, 2, \dots, m^i$), m^i 表示工位 i 中并行异速机数量.

2.1 基本假设

假设 1 每项任务在每个工位只选择一台设备加工一次.

假设 2 任一时刻每台设备只加工一项任务.

假设 3 每项任务按照相同的工艺路径依次通过各工位.

假设 4 任务在各工位一旦开始加工便不允许

收稿日期: 2010-06-13; 修回日期: 2010-09-02.

基金项目: 国家自然科学基金项目(70902037); 铁道部科技开发计划项目(2008Z019); 四川省教育厅项目(10ZC004).

作者简介: 杨琴(1978-), 女, 讲师, 博士生, 从事生产作业计划和优化调度等研究; 周国华(1966-), 男, 教授, 博士生导师, 从事生产管理等研究.

中断.

2.2 问题描述

采用三元组 α, β, γ 方法进行描述.

1) 资源环境 (α 域). 资源环境是柔性流水车间 (FFS), 包括 l 个串行工位, 每个工位包含多台并行机.

2) 任务特征和约束 (β 域). β 域包括多项内容, 本研究设置以下几项: ①到达时间约束 (r_j 表示任务 j 到达系统的时间); ②作业时间约束 (p_j^{ih} 表示任务 j 在工位 i 中机器 h 上的作业时间); ③交货期约束 (d_j 表示任务预期完成时间); ④机器使用限制 (M_j^i 表示工位 i 中可加工任务 j 的机器集合).

3) 调度目标 (γ 域). 调度目标的类别很多, 本研究设置目标函数为最小化加权滞后时间和 ($\sum \omega_j T_j$). 其中: T_j 表示任务滞后时间, $T_j = \max(C_j - d_j, 0)$, C_j 表示任务 j 完成所有作业的时间, C_j^i 表示任务 j 在工位 i 的完工时间; ω_j 表示任务的权重.

综上, 问题可描述为以最小化加权滞后时间和为目标, 受到达时间和机器使用限制的柔性流水车间动态调度问题 ($\text{FFS} | r_j, M_j | \sum \omega_j T_j$).

2.3 模型设计

$\text{FFS} | r_j, M_j | \sum \omega_j T_j$ 问题的数学模型如下:

$$\text{Min} \sum_{j=1}^n \omega_j T_j. \tag{1}$$

$$\text{s.t. } T_j = \max(d_j - C_j, 0), j \in N; \tag{2}$$

$$C_j = C_j^l, j \in N; \tag{3}$$

$$C_j^i - r_j^i \geq p_j^{ih}, i \in L, j \in N, h \in M^i; \tag{4}$$

$$x_j^{1h} \geq r_j, j \in N, h \in M^1; \tag{5}$$

$$x_j^{kh} - x_j^{ih} \geq p_j^{ih}, (i, j) \rightarrow (k, j) \in A; \tag{6}$$

$$x_j^{ih} - x_g^{ih} \geq p_g^{ih} \text{ or } x_g^{ih} - x_j^{ih} \geq p_j^{ih}, \tag{7}$$

$$i \in L, h \in M^i;$$

$$x_j^{ih} \geq 0, i \in L, j \in N, h \in M^i. \tag{8}$$

其中: 决策变量 x_j^{ih} 表示操作 (ih, j) 的起始时间; N 表示任务集合; L 表示工位集合, M^i 表示工位 i 中的机器集合; A 为线路约束 $(i, j) \rightarrow (k, j)$ 集合, 即任务 j 先在工位 i 加工, 接着在工位 k 加工. 式 (1) 为目标函数. 式 (2)~式 (8) 为约束条件. 其中: 式 (2) 度量任务 j 离开系统的时间; 式 (3) 确保任务 j 有足够时间完成工位 i 上的作业; 式 (4) 保证任务 j 能完成所有工序; 式 (5) 确保任务 j 在任务到达后才开始操作; 式 (6) 为操作约束, 确保任务 j 在不同工位的加工顺序; 式 (7) 为机器约束, 保证在同一机器上作业的不同任务间的先后顺序; 式 (8) 确定操作起始时间的取值范围.

3 算法设计

基于 DBR 理论的调度思路是将复杂的大规模问题聚焦到瓶颈. 首先松弛非瓶颈机的能力约束, 调度最难安排的瓶颈机; 接着安排非瓶颈机; 然后协调各子问题, 以获得一个好的可行调度.

3.1 问题分解及子问题建模

3.1.1 问题分解

根据 DBR 理论, 流水线上的设备可以分为瓶颈机和非瓶颈机, 非瓶颈机又包括瓶颈前工位和瓶颈后工位. 在瓶颈与非瓶颈工位间设置缓冲, 则 FFS 系统可分为 3 个部分, 如图 1 所示.

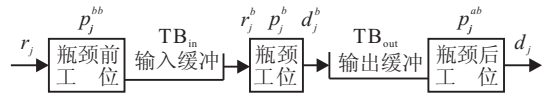


图 1 基于瓶颈的问题分解

图 1 中: $p_j^{bb} = \sum_{i=1}^{b-1} p_j^i$ 表示瓶颈前工位作业时间和,

TB_{in} 表示输入缓冲; $p_j^{ab} = \sum_{i=b+1}^l p_j^i$ 表示瓶颈后工位作

业时间和, TB_{out} 表示输出缓冲; $r_j^b (r_j^b = r_j + p_j^{bb} + \text{TB}_{in})$ 表示瓶颈工位到达时间; p_j^b 表示瓶颈工位作业时间, $d_j^b (d_j^b = d_j - p_j^{ab} - \text{TB}_{out})$ 表示瓶颈工位交货期.

3.1.2 子问题建模

通过设置缓冲确保瓶颈环节任务的连续性和及时性, 则可以松弛非瓶颈机的能力约束. FFS 系统 ($\text{FFS} | r_j, M_j | \sum \omega_j T_j$) 动态调度问题可简化为瓶颈工位并行异速机 ($Q_m | r_j, M_j | \sum \omega_j T_j$) 动态调度问题.

引理 1 令 X 和 X_b 分别表示 FFS 问题和瓶颈工位 Q_m 问题的最优解, 则有 $X_b \leq X$.

证明 令 Z 和 Z_b 表示 FFS 问题和瓶颈 Q_m 问题解的可行域, 松弛机器能力约束, 则有 $Z \subseteq Z_b$. 又因为目标函数相同且为最小化问题, 所以 $X_b \leq X$, 即瓶颈调度问题的最优解是原问题最优解的下界. \square

3.2 瓶颈和非瓶颈工位的动态调度

3.2.1 重调度时刻的确定

基于工件的混合重调度策略如图 2 所示.

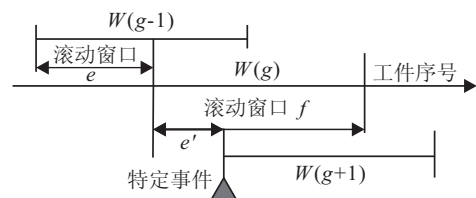


图 2 基于工件的混合重调度窗口

图 2 中: 横坐标表示工件序号, $W(g)$ 表示第 g 个预测窗口, 预测窗口内有 f 个工件. 重调度时刻有 2 种情况: 情况 1, 周期性重调度, 依据滚动窗口, 每次执行局部最优调度中的 $e(e < f)$ 个工件, 并在其完成时刻重调度; 情况 2, 特定事件(包括紧急工件到达、工件延迟、优先权变化等)发生时, 虽只完成了 $e'(e' \leq e)$ 个工件, 仍需重调度.

3.2.2 瓶颈工位调度

瓶颈工位采取完全重调度, 对未加工的所有工件集进行调度. 调度步骤如下:

1) 机器分配. 根据 Lfj(最低柔性工作优先) 规则和 LFM(最低柔性机器优先) 规则^[3], 将预测窗口中所有任务分配到各个机器. 出现不同任务和设备的柔性相同时, 随机分配, 形成多套方案.

2) 作业排序. 根据 ATC(明显滞后成本) 规则^[3], 决定每台设备上任务的加工顺序. 计算任务的指数, 选择指数大的优先调度.

3) 遗传算法求解. 将前阶段得到的调度方案用作遗传算法的初始解. 定义 $n \times 2$ 维矩阵

$$\begin{bmatrix} a_1 & a_2 & \cdots & a_n \\ b_1 & b_2 & \cdots & b_n \end{bmatrix}$$

作为遗传算法的染色体. 其中: a_k 表示设备, b_k 表示加工序号^[11]. 令适配值函数为目标函数的倒数, 通过交叉变异迭代搜索, 得到更好的调度.

3.2.3 非瓶颈工位调度

瓶颈前工位的任务到达时间 $r_j^{bb} = r_j$, 交货期 $d_j^{bb} = t_j^b - TB_{in}$; 瓶颈后工位的任务到达时间 $r_j^{ab} = c_j^b$, 交货期 $d_j^{ab} = d_j$. t_j^b 和 c_j^b 表示瓶颈工位调度结果的开始时间和完工时间.

非瓶颈工位采取部分重调度, 即仅重调度受干扰影响的任务. 瓶颈前工位采用 MS(最小松弛优先) 动态分派规则^[3] 作业排序, 瓶颈后工位采用 ERD(到达日期最早优先) 规则^[3] 作业排序.

3.3 瓶颈与非瓶颈工位间的协调

运用上述算法对瓶颈和非瓶颈工位进行预调度. 调度结果中, 任务 j 在瓶颈前工位作业的实际完成时间用 C_j^{bb} 表示.

1) 设置缓冲. 为确保瓶颈环节任务不间断, 任务能及时完成, 不拖延最终完工时间, 在瓶颈工位前设置输入缓冲 ($TB_{in} = 2P_j^{bb}$), 在瓶颈工位后设置输出缓冲 ($TB_{out} = P_j^{ab}$).

2) 缓冲管理. 如果瓶颈前工序不能在期望时间内完成, 则造成任务作业时间迭代, 使调度不可行. 所以需加强输入缓冲区的管理, 以缓解矛盾, 如图 3 所示. 忽略区: $C_j^{bb} \in [r_j + p_j^{bb}, r_j + 2p_j^{bb}]$, 落入此区域的任务在重调度时可忽略, 只需将新任务插入其中即可, 如果

有大量任务落入此区域, 则可考虑缩短输入缓冲; 警示区: $C_j^{bb} \in [r_j + 2p_j^{bb}, r_j^b]$, 应关注落入此区域的任务, 如果该任务在瓶颈工位的作业开始时间 t_j^b 变化, 则需重新计算其指数, 进行重排; 冻结区: $C_j^{bb} \in [r_j^b, t_j^b]$, 任务落入此区域则需冻结其在瓶颈前工位的作业时段, 确保该任务能按时执行, 如果有大量任务落入此区域, 则需延长各任务的交货期, 并重新设计输入缓冲; 重排区: $C_j^{bb} \geq t_j^b$, 对于落入此区域的任务, 需进行重新调度, 调整瓶颈工位的到达时间和完工时间, 并根据新的调度结果重新设定非瓶颈工位的到达时间和交货期.

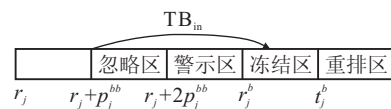


图 3 输入缓冲区管理

4 仿真测试

为了测试算法性能, 在文献 [12] 的基础上进行改进, 随机生成测试数据. 任务数量 $n \in \{30, 50, 80\}$, 任务的权重在区间 $[1, 3]$ 中正态分布, 其中可以使用辅助设备的任务占任务总数的 50%; 工位个数为 5, 瓶颈位于第 3 个工位; 瓶颈资源总量 $m \in \{3, 4, 5\}$; 瓶颈工位上任务的加工时间服从 $[1, 50]$ 均匀分布, 非瓶颈工位上任务的加工时间服从 $[0.3, 15]$ 均匀分布; 任务到达时间服从 $[0, \gamma E(C_{max})]$ 的离散均匀分布, $E(C_{max}) = nE(p_j^i)$, $\gamma = 0.2$; 任务交货期服从 $(\mu - \mu\theta/2, \mu + \mu\theta/2)$ 的离散均匀分布, $\mu = (1 - \tau)E(C_{max})$, 工件的拖期比率 $\tau = 0.6$, 交货期分布范围参数 $\theta = 1.5$.

由于动态 FFS 问题是强 NP 难题, 目前还没有精确算法能在短时间内获得最优解. 因此, 本文运用几种调度规则(先到先加工 FCFS, 最短加工时间先加工 SPT, 最早交货期限先加工 EDD) 进行完全重调度, 并将本文所述算法记作 DBR 进行仿真比较. 以 5 个任务为滚动窗口, 用 $n \times m[A(x), B(y)]$ 标志每个组别. 其中: n 为任务数量, m 为瓶颈设备的总量, A 和 B 分别表示瓶颈设备中的通用和辅助设备, 括号内的 x, y 表示对应资源的数量.

采用以下评价指标: 定义一组满足相同特征的问题 S , 令 $I(I \in S)$ 为符合问题特征的实例, $T(H, I)$ 为针对实例 I 采用启发式算法 H 求得的目标函数值, 则该实例在几种启发式算法下的最优解为 $BEST(I)$, 且 $BEST(I) = \min\{T(H, I)\}$. 据此, 对于每类问题 S , 可以计算相对性能指标如下:

$$\rho(H, S) = \frac{\sum_{I \in S} T(H, I)}{\sum_{I \in S} BEST(I)} \quad (9)$$

所有算法均采用 Matlab 编程, 在奔腾双核、主频 1.73 GHz 的计算机上进行仿真测试, 结果如表 1 所示。

表 1 调度性能比较

$n \times m[A(x), B(y)]$	调度性能参数 $\rho(H, S)$			
	DBR	FCFS	SPT	EDD
$30 \times 3[A(1), B(2)]$	1.000	1.376	1.136	1.080
$50 \times 3[A(1), B(2)]$	1.000	1.485	1.240	1.112
$50 \times 4[A(2), B(2)]$	1.000	1.567	1.223	1.119
$80 \times 4[A(2), B(2)]$	1.000	1.798	1.462	1.257
$80 \times 5[A(2), B(3)]$	1.000	1.902	1.585	1.226

由表 1 数据可得以下结论: 基于 DBR 理论调度算法的仿真结果显著优越, 且随问题规模的增大, 指标性能的优越性更为明显。

5 结 论

FFS 动态调度的复杂性以及对实际问题的符合性, 表明了对其进行研究具有重要的理论和现实意义。结合 DBR 理论, 可使问题简化, 在此基础上研究如何运用重调度策略和启发式算法实现该问题的求解, 并通过仿真实例验证了所提出算法的可行性和有效性。

参考文献(References)

- [1] Zhang C S, Sun J G, Zhu X J. An improved particle swarm optimization algorithm for flow-shop scheduling Problem[J]. Information Processing Letters, 2008, 108(4): 204-209.
- [2] Wang H. Flexible flow shop scheduling optimum heuristics and artificial intelligence solutions[J]. Expert Systems, 2005, 22(2): 78-85.
- [3] Pinedo M. Scheduling: Theory, algorithms, and systems[M]. 2nd ed. New Jersey: Prentice Hall, 2001.
- [4] Sun J, Xue D. A dynamic reactive scheduling mechanism for responding to changes of production orders and manufacturing resources[J]. Computers in Industry, 2001, 46(2): 189-207.
- [5] Hédi Chtourou, Mohamed Haouari. A two stage priority rule based algorithm for robust resource constrained project scheduling[J]. Computers & Industrial Engineering, 2008, 55(8): 183-194.
- [6] Goldratt E M, Cox J. The goal: A process of ongoing improvement[M]. NY: North River Press, 1984.
- [7] 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.)
- [8] 郭永辉, 钱省三. 基于鼓-缓冲器-绳子理论的整合式生产作业控制系统研究[J]. 计算机集成制造系统, 2006, 12(2): 252-256. (Guo Y H, Qian X S. Integrated production planning and control system based on drum buffer rope theory[J]. Computer Integrated Manufacturing Systems, 2006, 12(2): 252-256.)
- [9] Betterton C E, Cox III J F. Espoused drum-buffer-rope flow control in serial lines: A comparative study of simulation models[J]. Int J of Production Economics, 2009, 1(1): 66-79.
- [10] Chen C L, Chen C L. A bottleneck-based heuristic for minimizing makespan in a flexible flow line with unrelated parallel machines[J]. Computers & Operations Research, 2009, 36(2): 3073-3081.
- [11] 刘民, 吴澄, 尹文龙. 带特殊工艺约束的并行机器生产线调度问题的一种遗传算法[J]. 自动化学报, 2001, 27(3): 381-386. (Liu M, Wu C, Yin W L. Solving identical parallel machine production line scheduling problem with special procedure constraint by genetic algorithm[J]. Acta Automatica Sinica, 2001, 27(3): 381-386.)
- [12] Lee G C, Kim Y D, Choi S W. Bottleneck focused scheduling for a hybrid flow shop[J]. Int J of Production Research, 2004, 42(11): 165-181.
- [10] 张晓伟, 刘三阳. 免比例因子 F 的差分进化算法[J]. 电子学报, 2009, 36(6): 1318-1323. (Zhang X W, Liu S Y. Differential evolution without the scale factor F [J]. Acta Electronica Sinica, 2009, 36(6): 1318-1323.)
- [11] 戈剑武, 祁荣宾, 钱锋. 一种改进的自适应差分进化算法[J]. 华东理工大学学报: 自然科学版, 2009, 35(4): 600-605 (Ge J W, Qi R B, Qian F. A modified adaptive differential evolution algorithm[J]. J of East China University of Science and Technology: Natural Science Edition, 2009, 35(4): 600-605.)
- [12] Qin Suganthan. Self adaptive differential evolution algorithm for numerical optimization[C]. Proc of the IEEE Congress on Evolutionary Computation. Edinburgh: Institute of Electrical and Electronics Engineers Computer Society, 2005: 1785-1791.
- [13] David W Hosmer, Stanley Lemeshow. Applied logistic regression[M]. 2nd ed. New York: Wiley-Interscience Publication, 2000.

(上接第 1108 页)