

基于 Petri 网的 JSP 动态优化调度

陶泽^{1,2}, 徐炜达¹, 肖田元¹, 郝长中²

(1. 清华大学自动化系, 北京 100084; 2. 沈阳理工大学机械工程学院, 沈阳 110168)

摘要: 提出一种在柔性制造系统动态优化调度中处理紧急订单的方法。以带有控制器的 Petri 网为建模工具对柔性生产调度中的离散事件建模, 对系统的设备维护、各种优先级等特性进行描述, 利用遗传算法和模拟退火算法获得调度结果, 用于解决作业车间的加工受到机床、操作工人等双资源约束条件下的动态优化调度。当有紧急订单需要加工时, 该方法把剩余任务和紧急任务作为两个独立的任务分别处理, 然后进行集成, 在紧急任务为最优调度的基础上选取剩余任务的最优调度, 找到兼顾整体和局部的最优解。仿真结果说明了算法的有效性和鲁棒性。

关键词: Petri 网; 控制器; 动态车间调度; 紧急订单

Dynamic JSP Scheduling Based on Petri Net

TAO Ze^{1,2}, XU Wei-da¹, XIAO Tian-yuan¹, HAO Chang-zhong²

(1. Department of Automation, Tsinghua University, Beijing 100084;

2. School of Mechanical Engineering, Shenyang Ligong University, Shenyang 110168)

【Abstract】 A new method of disposing urgent order is presented. A Petri net with controller is used to model discrete events in flexible job shop scheduling, it not only has the modeling capability of a traditional Petri net, but also can depict system characteristics, such as equipment maintenance, different types of priorities, and so on. The scheduling results are obtained based on genetic and simulated annealing algorithm. The method is developed to address the dynamic scheduling problem in manufacturing systems constrained by machines, workers. When urgent order comes, remainder jobs and urgent jobs are disposed separately, then being integrated, and through finding optimal results of remainder jobs based on urgent jobs result is optimal, and the whole and local optimal results can be found through this method. Simulation results based on a job shop scheduling show that the proposed method is efficient and robust.

【Key words】 Petri net; controller; dynamic job shop scheduling; urgent order

具有柔性加工路径的 Job Shop 型调度问题 (JSP) 是一个具有 NP 难计算复杂性的开放问题。针对调度问题研究人员已经提出了很多方法, 例如排队论模型、数学规划、系统仿真、控制理论等, 但是这些方法描述问题的能力很有限, 所以寻找更加有效的方法是很有必要的。由于 Petri 网具有直观的图形表示和严密的数学基础, 因此 Petri 网作为离散事件动态系统的建模和分析工具, 已被成功地应用于柔性生产系统的建模分析和控制^[1-2]。本文应用具有库所和变迁混合约束的 Petri 网控制器设计方法^[3]进行控制器的设计, 并用 Petri 网进行建模; 在此 Petri 网模型基础之上利用遗传算法和模拟退火算法 (GASA)^[4]进行优化, 通过对该 Petri 网模型的仿真运行可以证实该调度策略具有良好的优化性能。

1 柔性生产调度的 Petri 网模型

1.1 Petri 网控制器的设计

设计 Petri 网控制器的目的就是使系统的运行满足期望的目标, 避免禁止状态的发生。

一般把式 (1) 中的库所和变迁称为约束库所和约束变迁, 以 C 表示, 即

$$C = \left\{ \sum_{i=1}^n l_i M(p_i) + \sum_{i=1}^m \lambda_i q_i, b \right\} \quad l_i \quad 0, \lambda_i \in \{0, 1\} \quad (1)$$

其中, q_i 是第 i 个变迁的激发索引, 且 $\lambda_i = 1$ 表示第 i 个变迁激发了。设计步骤如下:

步骤 1 确定约束库所 C_p 和约束变迁 C_t ;

步骤 2 在约束库所的融合库所 p_e ^[3] 的基础上, 求得库所 p_{c0} , 其关联矩阵 $P_{c0} = -P_e$; 即

$$P_{c0} = p_{c0} \begin{bmatrix} t_1 & t_2 & \cdots & t_j \\ -\sum_{i=1}^k l_i w_{i1} & -\sum_{i=1}^k l_i w_{i2} & \cdots & -\sum_{i=1}^k l_i w_{ij} \end{bmatrix} \quad (2)$$

根据 P_{c0} 构建 p_{c0} 与对应变迁之间弧的连接方式。

步骤 3 对于 $t \in C_t$, 在矩阵 P_{c0} 的下方对应填入约束变迁的位置上写 1, 其他位置写 0 (为了便于理解假设 $\lambda_2 = 0$), 即

$$P_c = p_c \begin{bmatrix} t_1 & t_2 & \cdots & t_j \\ -\sum_{i=1}^k l_i w_{i1} & -\sum_{i=1}^k l_i w_{i2} & \cdots & -\sum_{i=1}^k l_i w_{ij} \\ 1 & 0 & & 1 \end{bmatrix} \quad (3)$$

(1) 如果末行元素 1 对应的矩阵中的元素为 0, 在控制器库所 p_c 和该变迁之间构建一条双向弧, 即让 t 既是 p_c 的输入变迁又是输出变迁。

(2) 如果末行元素 1 对应的矩阵中的某元素小于 0, 在 p_c 和 t 之间无需构建额外的弧。

基金项目: 科技部中小企业创新基金资助项目 (05C26211400246)

作者简介: 陶泽 (1977 -), 女, 博士后, 主研方向: 车间调度, Petri 网及性能分析; 徐炜达, 博士研究生; 肖田元, 教授、博士生导师; 郝长中, 教授、博士

收稿日期: 2007-01-24 **E-mail:** taoze@mail.tsinghua.edu.cn

(3)如果末行元素 1 对应的矩阵中大于 0 的某元素对应的变迁与末行元素 0 对应的矩阵中大于 0 的某元素对应的变迁有共同的输入库所,则在控制器库所 p_c 和该变迁之间构建一条双向弧,否则控制器不能用此方法来设计,因为控制器可能会导致死锁。

步骤 4 如果控制器存在它的初始标识为

$$M_0(p_c) = b - \sum_{i=1}^k l_i M_0(p_i) \quad (4)$$

由于该控制器设计方法只需要考虑直接或间接的与约束有关的变迁,因而比Yang^[5]提出的 Petri 网控制器设计方法更加简单、有效并且不会导致死锁,对于复杂控制系统,其优势更加明显。

1.2 柔性生产调度的 Petri 网模型

为了直观地说明柔性生产调度的 Petri 网模型,建立一子模型。考虑如下的一个 FMS 系统,其包括 3 种类型 6 台机器,1,2 号机器为类型 1;3 号机器为类型 2;4,5,6 号机器为类型 3。共有 10 个工件需要加工,每个工件有 1~4 个工序。工人工作表、任务参数表如表 1 和表 2 所示。工人只是第二资源的典型代表,根据实际情况也可以是机器人等其他重要的资源。

表 1 工人与机床设备工作表

工人	机床 1	机床 2	机床 3	机床 4	机床 5	机床 6
1	操作	操作	—	—	—	—
2	—	操作	操作	—	—	—
3	—	—	—	操作	操作	—
4	—	—	—	—	操作	操作

表 2 任务参数表

工件	工序	加工时间	机器类型	工件	工序	加工时间	机器类型
1	1	12	—	5	2	10	—
	2	9	—	—	3	10	—
	3	5	—	—	4	8	—
2	1	6	—	6	1	6	—
	2	6	—	7	1	6	—
3	3	8	—	—	2	8	—
	1	5	—	8	1	12	—
4	2	9	—	—	2	8	—
	1	4	—	—	3	8	—
5	2	6	—	9	1	5	—
	3	10	—	—	2	8	—
6	1	5	—	10	1	7	—

建立工件 1 的 Petri 模型如图 1 所示,其中, p_i 表示第 i 个工件的初始状态; p_i^m 表示第 i 台机器; $p_{i,j,h,k}$ 表示第 i 个工件的第 j 个操作由第 h 个工人在第 k 台机器上进行; $t_{i,j,h,k}^s$ 表示 $p_{i,j,h,k}$ 操作开始; $t_{i,j,h,k}^e$ 表示 $p_{i,j,h,k}$ 操作结束; $p_{i,j}^b$ 表示工件 i 的第 j 个操作结束后的缓冲区; p_i^f 表示第 i 个工件加工完成。虚线部分表示控制器部分,主要对以下两种行为进行控制:

(1)第 1 个缓冲区不能上溢也不能下溢,当缓冲区内有工件时机器 3 不能启动操作,即 $m(p_{1,1,2,3}) + m(p_{1,1}^b) = 1$ 。

(2)机器 3 有比机器 1 修复和回到正常工作的优先级。按照此方法可以建立其余工件的 Petri 网模型,最后,将这些模型通过表示机器与工人的库所联结起来,便得到系统的整个模型。

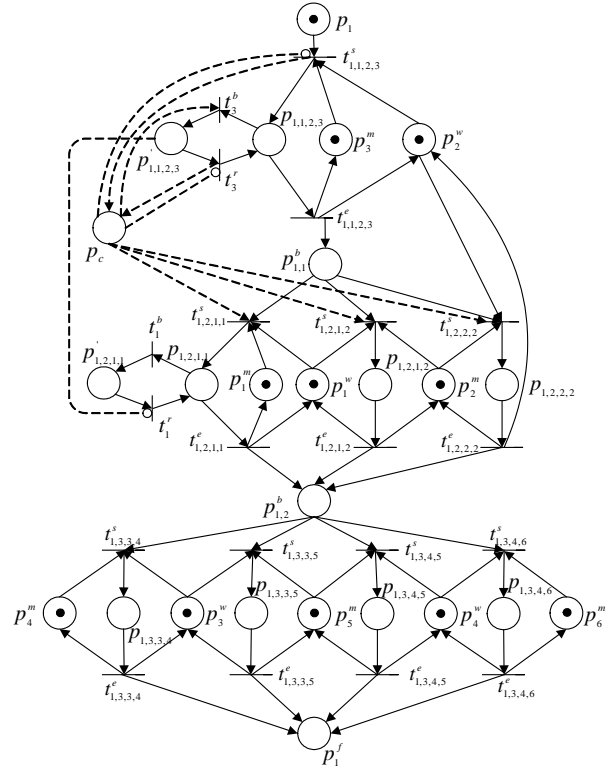


图 1 零件 1 的 Petri 网模型

2 调度数学模型的建立

研究的调度模型是:作业车间需要加工多种工件,每种工件有多条工艺加工路线;要求制定出一个生产计划,它不仅为每一个工件决定一条工艺加工路线,而且还必须满足整个车间的生产周期最短。

调度的目标为

$$\min Z \quad (5)$$

主要约束条件如下:

(1)零件 i 的第 j 条工艺加工路线中的最后一道工序

$$T_{ijhm} = Z \quad (6)$$

(2)零件 i 的第 j 条工艺加工路线中的非最后一道工序

$$T_{ijhm} - T_{ij(h-1)g} = t_{ijhm} \quad (7)$$

$$\forall i, j, m, h, g; h \neq 1$$

(3)零件 i 的第 j 条工艺加工路线和零件 p 的第 q 条工艺加工路线中都有工序要在机床设备 m 上加工

$$T_{ijhm} - T_{pqsm} + HY_{ijhpqsm} = t_{ijhm} \quad (8)$$

(4)零件 i 的第 j 条工艺加工路线和零件 p 的第 q 条工艺加工路线中都有工序由工人 w 来加工

$$T_{ijhw} - T_{pqsw} + HY_{ijhpqsw} = t_{ijhw} \quad (9)$$

式中, Z 为生产周期; i 为零件; j 为属于一个零件的一条工艺加工路线; m 为机床设备; h 为零件 i 的某一条工艺加工路线的第 h 道工序; t_{ijhm}, T_{ijhm} 分别为零件 i 的第 j 条工艺加工路线中的第 h 道工序在机床 m 上的加工时间和加工完毕时刻; H 为非常大的正数; w 为工人; t_{ijhw}, T_{ijhw} 分别为零件 i 的第 j 条工艺加工路线中的第 h 道工序由工人 w 加工的时间和加工完毕的时刻; $Y_{ijhpqsm}$ 为机床 m 加工工序 h 和 s 的顺序

判别条件。

当工序 h 和 s 都在机床 m 上被加工时, 如果零件 i 的第 j 条工艺加工路线中的第 h 道工序先于零件 p 的第 q 条工艺加工路线中的第 s 道工序被加工, 则 $Y_{ijhpqsm} = 1$, 否则 $Y_{ijhpqsm} = 0$; $Y_{ijhpqsw}$ 为工人 w 加工工序 h 和 s 的顺序判别条件, 当工序 h 和 s 都由工人 w 加工时, 如果零件 i 的第 j 条工艺加工路线中的第 h 道工序先于零件 p 的第 q 条工艺加工路线中的第 s 道工序被加工, 则 $Y_{ijhpqsw} = 1$, 否则 $Y_{ijhpqsw} = 0$ 。

3 紧急定单的处理

紧急定单是指交货期提前或急需加工的定单。在系统的实际加工过程中, 紧急定单所包含的工件(紧急工件)比其他工件有较高的优先权, 所以在调度的时候要优先考虑紧急工件。但是当 $\min Z_2 < \min Z_1$ 时(其中 Z_1 为紧急任务与剩余任务组成的新任务的生产周期, Z_2 为剩余任务的生产周期), 使用传统方法得到的整体调度结果虽然没错, 但局部调度并不一定是最优的。

在这里提出一种新的策略用来解决紧急工件的调度, 该方法可以有效避免上述现象的发生。把紧急工件的加工和其他剩余工件的加工当作两个不同的独立任务来处理。首先, 先处理紧急工件, 然后再处理其他工件; 虽然是两个不同的任务但是剩余工件的处理不能在紧急工件占用的时间段上(剩余工件和紧急工件由同一工人或在一台机床上加工时); 在紧急工件目标值最优的基础上选取剩余工件目标值最优的调度。

这种策略与上述传统的调度整体结果虽然是相同的, 但是局部比其更优。

$$\min Z_1 \quad (10)$$

$$\min Z_1 \cap \min Z_2 \quad (11)$$

传统的处理紧急工件的方法满足式(10), 而本文提出的新方法满足式(11)。

4 仿真结果与分析

对表 1 和表 2 所示的例子根据 Petri 网描述的加工路径和资源约束应用混合遗传算法和模拟退火算法进行仿真运算。种群规模 50, 交叉率为 0.8, 变异率为 0.1, 退温指数 $\lambda = 0.9$, 进化 50 代后得到的调度方案见图 2, 目标函数值为 55。

图 2 调度 Gantt 图显示了工序和机床的分配关系。不同的零件及工序用 4 个字符在 Gantt 图上来标识, 前两位表示零件号, 第 3 位表示工序号, 第 4 位表示工人; 例如第 1 台

机床上的“0511”表示第 5 个工件的第 1 道工序由第 1 个工人加工。工人任务分配图显示了零件与操作工人和机床的对应关系, 图中纵坐标表明了操作工人。从这张图中, 也可以很容易地统计各个工人的工作量, 比如, 工人 1 的工作量是 54 (5+12+4+5+10+10+8), 工人 2、工人 3 和工人 4 的工作量分别是 55、34 和 38。

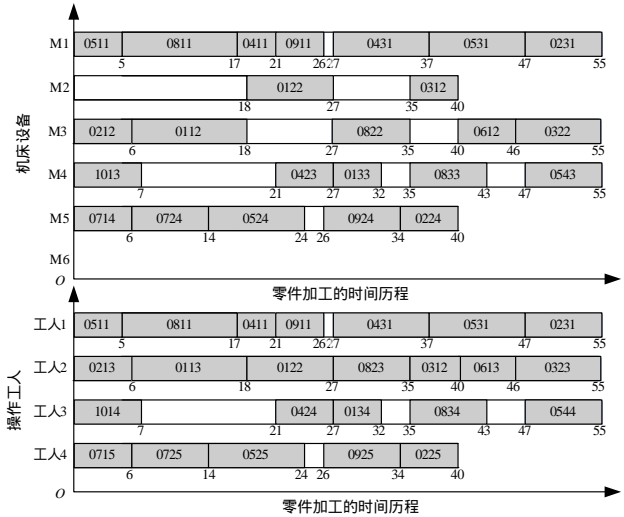


图 2 静态调度 Gantt 图

如果在 $t=33$ 时有紧急定单需要加工(见表 3), 调度结果见图 3。其中 Gantt 图中的竖实线前的部分为紧急定单未到时的静态调度, 黑色部分为紧急工件的加工, 图 3 为采用传统方法得到的 Gantt 图, 图 4 为采用本文提出的新方法得到的调度结果 Gantt 图。虽然两种调度结果的目标值相同都为 104, 但在调度结果上如果不考虑紧急工件, 即黑色部分, 传统方法和新方法的目标值分别为 80 和 76, 说明新方法优于传统方法。

表 3 紧急定单加工任务

工件	工序	加工时间	机器类型
11	1	5	
	2	10	
	3	6	
12	1	10	
	2	6	
13	1	8	
	2	8	
	3	10	

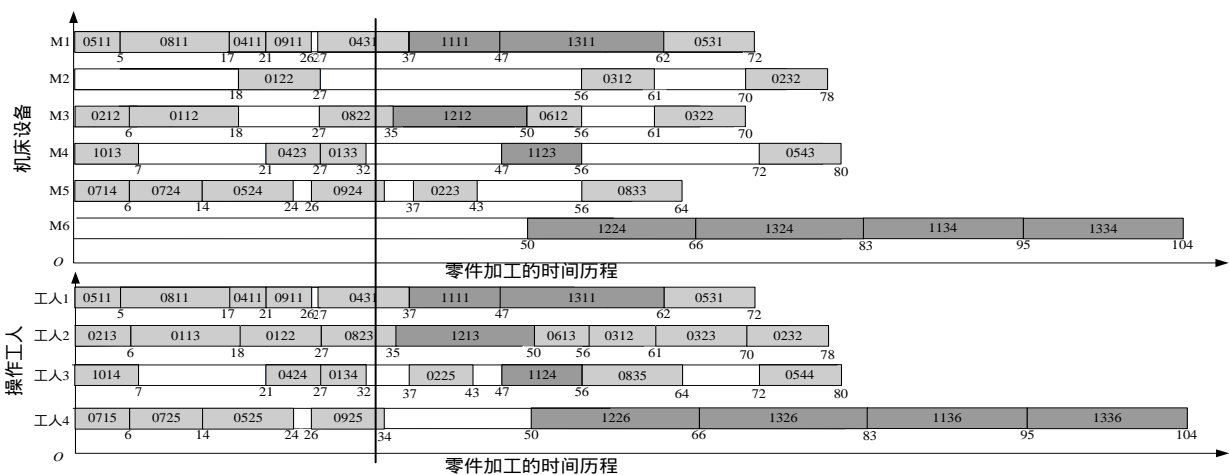


图 3 传统的紧急事件调度 Gantt 图

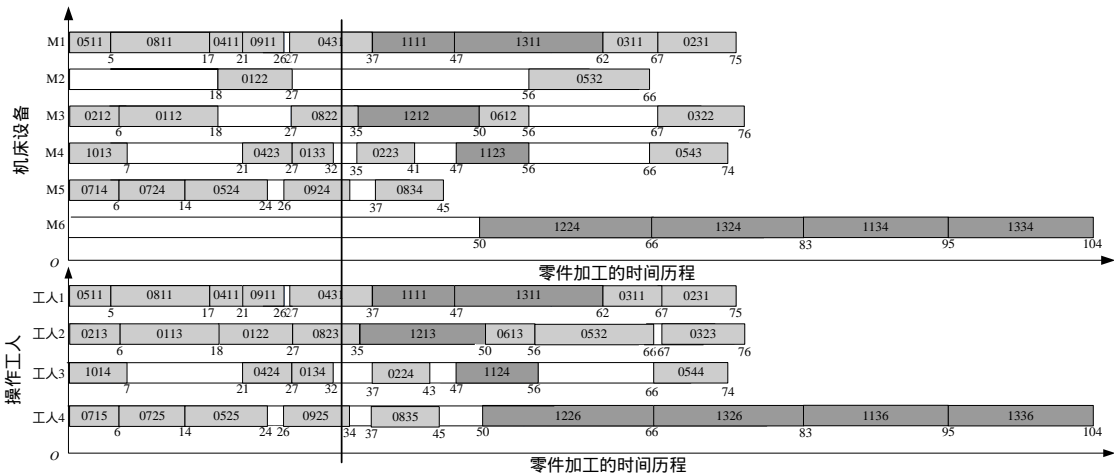


图4 新方法的紧急工件调度 Gantt 图

5 结束语

本文针对 Job Shop 调度问题,建立了带有控制器的 Petri 网模型,该模型不仅能够反映柔性生产线的待加工产品的多条加工路径及其资源约束,还可以对系统的设备维护、各种优先级等特性进行描述,得到对生产系统更完善更精确的刻画。利用 Petri 网的广度优先遍历与 GASA 结合而得到的调度算法,可以完成双资源调度问题的动态求解。提出的处理紧急定单的新方法能够找到兼顾整体和局部的最优解,有效地避免了传统方法存在的缺点,该算法还可以扩展到多资源 JSP 优化调度。

参考文献

[1] Chen Jyh-horng, Fu Li-chen, LIN Ming-hung, et al. Petri-net and

GA-based Approach to Modeling, Scheduling, and Performance Evaluation for Wafer fabrication[J]. IEEE Trans. on Robotics and Automation, 2001, 17(5): 619.

[2] 王芙蓉, 吴铁军. 基于 Petri 网仿真的柔性生产调度——蚁群-遗传传递进化优化方法[J]. 浙江大学学报, 2004, 38(3): 286-291.
 [3] Tao Ze, Xie Li-yang, Liang Di. Controller Design of DES Petri Nets with Mixed Constraint[J]. Chinese Journal of Aeronautics, 2005, 18(3): 283-288.
 [4] 王 凌. 智能优化算法及其应用[M]. 北京: 清华大学出版社, 2001.
 [5] Yang Y Y, Linkens D A. Design of Petri Net Controllers to Exclude Forbidden States in Manufacturing Systems[C]//Proc. of the 14th World Congress of IFAC. Beijing: [s. n.], 1999: 379-384.

(上接第 13 页)

表3 3种调度算法性能比较

算法	调度长度	处理机数	加速比	通信计算比	算法时间复杂度
OSA 调度法	32	4	1.34	0.18	$O(V ^2)$
DAG 重构调度法	32	3	1.34	0.18	$O(\log V \cdot (V + E))$
DTO 调度法	30	3	1.43	0.08	$O(\log V \cdot (V + E))$

其中,

$$\text{加速比 } Speedup = \frac{\text{串行执行时间}}{\text{并行调度执行时间}} \quad (10)$$

$$\text{通信计算比 } CCR = \frac{\text{平均通信时间}}{\text{平均计算时间}} \quad (11)$$

由表3可以看出,除了在调度长度和处理机数目上得到显著优化外,DTO 调度算法还在加速比和通信计算比上较 OSA 调度法和 DAG 重构调度法有明显的优势。

另外,由于 $\log |V| < |V|$, 因此,相比其他两个算法,DTO 算法没有增加时间复杂度。

6 结束语

任务调度算法是并行计算系统的关键,本文分析了同层任务结点与其后继结点间的相关性,以使当前任务的直接后继具有以最早起始执行时间为主要目标,以减少处理机数目为次要目标,把任务尽量分配到已分配的处理机上的性能。

提出了一种基于相关任务图优化的任务调度算法 DTO,性能分析结果表明,该算法在调度长度、处理机数目和算法复杂度方面,要明显优于一些已有算法,对嵌入式并行系统中的多任务调度是一个较好的选择。

参考文献

[1] Garey M R, Johnson D S. Computers and Intractability: A Guide to the Theory of NP-completeness[M]. New York: W. H. Freeman and Co., 1979.
 [2] Adam T L, Chandy K, Dickson J. A Comparison of List Scheduling for Parallel Processing Systems[J]. Communications of the ACM, 1974, 17(12): 685-690.
 [3] Coffman E G. Computer and Job-shop Scheduling Theory[M]. New York: Wiley, 1976.
 [4] Park Chan-ik, Choe Tee-Young. An Optimal Scheduling Algorithm Based on Task Duplication[J]. IEEE Trans. on Computers, 2002, 51(4): 444-448.
 [5] Darbha S, Agrawal D P. Optimal Scheduling Algorithm for Distributed-memory Machines[J]. IEEE Trans. on Parallel and Distributed Systems, 1998, 9(1): 87-95.
 [6] 周佳祥, 郑纬民. 基于 DAG 图解——重构的机群系统静态调度算法[J]. 软件学报, 2000, 11(8): 1097-1104.