文章编号: 1001-0920(2001)0S-0697-03

# 任务具有链约束的平行机调度问题

赵传立[1], 张庆灵[1], 唐恒永[2]
(1.东北大学 理学院,辽宁 沈阳 110004;2.沈阳师范学院 数学系,辽宁 沈阳 110034)

摘　要: 研究任务间具有链约束的平行机调度问题,目标是在满足任务间链约束的条件下任务的总完工时间最小,这类问题是NP-难的。通过对问题的分析,对于一般情况给出了最优解的必要条件,对于特殊情况给出了问题的最优解算法。

关键词: 调度; 总完工时间; 链
中图分类号: O 223　　文献标识码: A

# Parallel Processor Scheduling with Chain-structured Tasks

ZHAO Chuan-li[1], ZHANG Qing-ling[1], TANG Heng-yong[2]
(1. College of Science, Northeastern University, Shenyang 110004, China; 2. Department of Mathematics, Shenyang Normal University, Shenyang 110034, China)

**Abstract**: A parallel processor scheduling problem with chain-structured tasks is studied. The objective is to minimize the total completion time. The problem is NP-hard. Through analyzing the problem, a necessary condition of optimal solution is given for general case. The optimal algorithms are presented for some special cases.

**Key words**: scheduling problems; total completion time; chains

## 1 引　　言

调度问题也称排序问题,所研究的问题是把一些任务安排在一些处理机上加工,使某个目标函数达到最小。通常的目标函数是最大完工时间和总完工时间。对于平行机调度问题,如果目标函数是最大完工时间,即使任务之间不存在优先约束,问题也是NP-难的[1],常用的求解方法是启发式算法;如果目标函数是总完工时间,在任务无优先约束时,SPT(shortest processing time first)规则产生最优调度方案,但任务间有某种优先约束时,问题是NP-难的,即使优先约束为最简单的链形约束,也是如此[2]。

本文研究任务的加工不允许中断,任务间具有链形约束的平行机调度问题,目标函数是极小化总完工时间。

## 2 问题模型及一般结论

问题的一般描述为:设有 $m$ 个处理机 $P_1, P_2, \cdots, P_m$, $n$ 个任务,任务加工不可中断,任务间具有链形优先约束,加工时链不可中断,即同一个链的任务必须连续加工,目标函数为总完工时间。该问题用三参数表示法可记为

$$Pm|\text{chains}|\sum C_j \qquad (1)$$

收稿日期: 2001-01-18
基金项目: 辽宁省科委自然科学基金项目(99107001);辽宁省教育厅科研基金项目(20262250,991121558)
作者简介: 赵传立(1958—),男,黑龙江泰来人,教授,博士生,从事控制与优化的研究;张庆灵(1956—),男,辽宁营口人,教授,博士生导师,从事广义大系统、容错控制等研究。

问题(1)是NP-难的, 对一般情况没有多项式最优算法。

为讨论方便,下面设 $n$ 个任务形成 $k$ 个平行链

$$L_1: T_{11} \to T_{12} \to \cdots \to T_{1n_1}$$

$$L_2: T_{21} \to T_{22} \to \cdots \to T_{2n_2}$$

$$\vdots$$

$$L_k: T_{k1} \to T_{k2} \to \cdots \to T_{kn_k}$$

第 $l$ 个链 $L_l$ 含有 $n_l$ 个任务, $l = 1, 2, \cdots, k$, $\sum_{l=1}^{k} n_l = n$。任务 $T_{ij}$ 的加工时间为 $p_{ij}$, $i = 1, 2, \cdots, k$; $j = 1, 2, \cdots, n_i$。第 $l$ 个链 $L_l$ 的加工时间记为

$$p_l = \sum_{j=1}^{n_l} p_{lj}, \quad l = 1, 2, \cdots, k$$

对于多个处理机的调度方案,可先将各个任务链列为一个表。每当有处理机可以加工任务时,则在表中从前到后依次取任务链排在处理机上。因此一旦列表完毕,则调度方案就已确定。如果表中链 $L_i$ 排在链 $L_j$ 前面,则在对应的调度方案中,链 $L_i$ 也在链 $L_j$ 之前加工,当然两个链可能分别在不同的处理机上加工。

下面先给出与问题(1)对应的单机调度问题的一个结论[3]:

引理 **1** 对于单机问题

$$1|\text{chains}|\sum C_j$$

把各个链按 $p_l/n_l$ 不减排列,即得最优调度方案。

对于 $m > 1$ 的情况,由于问题(1)是NP-难的,因此引理1的结论并不成立,但可有如下结论:

定理 **1** 设链 $L_i$ 由任务 $T_{i1}, T_{i2}, \cdots, T_{in_i}$ 组成,链 $L_j$ 由任务 $T_{j1}, T_{j2}, \cdots, T_{jn_j}$ 组成,若链 $L_i$ 与链 $L_j$ 满足: 1) $n_i > n_j$ 且 $p_i \le p_j$; 2) $n_i \ge n_j$ 且 $p_i < p_j$ 的条件之一。则在最优调度方案中链 $L_i$ 在链 $L_j$ 之前加工。

证明 这里用反证法证明链 $L_i$ 与链 $L_j$ 满足条件 1) 的情况,另一种情况同理可证。

设链 $L_i$ 与链 $L_j$ 满足 $n_i > n_j$ 且 $p_i \le p_j$,但在某个最优调度方案 $\pi$ 中,链 $L_i$ 排在链 $L_j$ 后面加工。下面证明存在某个调度方案,其目标函数值比最优调度方案 $\pi$ 的目标函数值还小,因此矛盾。

若在最优调度方案 $\pi$ 中,链 $L_i$ 与链 $L_j$ 排在同一处理机上,则由引理1,链 $L_i$ 应在链 $L_j$ 之前加工。

设在最优调度方案 $\pi$ 中,链 $L_i$ 排在处理机 $P_h$ 上,它的开始加工时间为 $t_1$; 处理机 $P_h$ 上排在链 $L_i$ 后面加工的任务有 $m_1$ 个,记为 $T_{h1}, T_{h2}, \cdots, T_{hm_1}$,其加工时间分别为 $p_{h1}, p_{h2}, \cdots, p_{hm_1}$; 链 $L_j$ 排在处理机 $P_s$ 上,它的开始加工时间为 $t_2$; 处理机 $P_s$ 上排在链 $L_j$ 后面加工的任务有 $m_2$ 个,记为 $T_{s1}, T_{s2}, \cdots, T_{sm_2}$,其加工时间分别为 $p_{s1}, p_{s2}, \cdots, p_{sm_2}$。

反证假设 $t_1 > t_2$, $n_i > n_j$ 且 $p_i \le p_j$。以下分两种情况证明存在某调度方案,其目标函数比最优调度方案 $\pi$ 更小。

1) $m_1 < m_2$, 交换链 $L_i$ 与链 $L_j$ 的位置,其余链位置不变,由此得到调度方案 $\pi_1$。

$\pi$ 与 $\pi_1$ 比较,只有链 $L_i$ 上任务及任务 $T_{h1}, T_{h2}, \cdots, T_{hm_1}$ 与链 $L_j$ 上任务及任务 $T_{s1}, T_{s2}, \cdots, T_{sm_2}$ 的完工时间不同。

在 $\pi$ 中处理机 $P_h$ 上,链 $L_i$ 上任务与任务 $T_{h1}, T_{h2}, \cdots, T_{hm_2}$ 的完工时间和为

$$n_i t_1 + \sum_{k=1}^{n_i} (n_i - k + 1) p_{jk} + m_1(t_1 + p_i) + \sum_{k=1}^{m_1} (m_1 - k + 1) p_{hk} \qquad (2)$$

在处理机 $P_s$ 上,链 $L_j$ 上任务与任务 $T_{s1}, T_{s2}, \cdots, T_{sm_2}$ 的完工时间和为

$$n_j t_2 + \sum_{k=1}^{n_j} (n_j - k + 1) p_{jk} + m_2(t_2 + p_j) + \sum_{k=1}^{m_2} (m_2 - k + 1) p_{sk} \qquad (3)$$

在 $\pi_1$ 中处理机 $P_h$ 上,链 $L_j$ 上任务与任务 $T_{h1}, T_{h2}, \cdots, T_{hm_1}$ 的完工时间和为

$$n_j t_1 + \sum_{k=1}^{n_j} (n_j - k + 1) p_{jk} + m_1(t_1 + p_j) + \sum_{k=1}^{m_1} (m_1 - k + 1) p_{hk} \qquad (4)$$

在处理机 $P_s$ 上,链 $L_i$ 上任务与任务 $T_{s1}, T_{s2}, \cdots, T_{sm_2}$ 的完工时间和为

$$n_i t_2 + \sum_{k=1}^{n_i} (n_i - k + 1) p_{ik} + m_2(t_2 + p_i) + \sum_{k=1}^{m_2} (m_2 - k + 1) p_{sk} \qquad (5)$$

比较式(2) ~ (5)可知, $\pi$ 的总完工时间比 $\pi_1$ 的总完工时间多

$$(n_i - n_j)(t_1 - t_2) + (m_2 - m_1)(p_j - p_i) \qquad (6)$$

由于 $n_i > n_j$, $t_1 > t_2$, $m_2 > m_1$, $p_j \ge p_i$, 因此式(6)大于0,与 $\pi$ 是最优调度矛盾。

2) $m_1 \ge m_2$, 交换链 $L_i$, 任务 $T_{h1}, T_{h2}, \cdots, T_{hm_1}$ 与链 $L_j$, 任务 $T_{s1}, T_{s2}, \cdots, T_{sm_2}$ 的位置,其余链位置不多,由此得到一个调度方案 $\pi_2$。

$\pi_2$ 与 $\pi_1$ 比较, 只有链 $L_i$ 上任务及任务 $T_{h1}, T_{h2}, \cdots, T_{hm_1}$ 与链 $L_j$ 上任务及任务 $T_{s1}, T_{s2}, \cdots, T_{sm_2}$ 的完工时间不同。

在 $\pi_2$ 中处理机 $P_h$ 上, 链 $L_j$ 上任务与任务 $T_{s1}, T_{s2}, \cdots, T_{sm_2}$ 的完工时间和为

$$n_j t_1 + \sum_{k=1}^{n_j} (n_j - k + 1) p_{jk} + m_2 (t_1 + p_j) + \sum_{k=1}^{m_2} (m_2 - k + 1) p_{sk} \quad (7)$$

在处理机 $P_s$ 上, 链 $L_i$ 上任务与任务 $T_{h1}, T_{h2}, \cdots, T_{hm_1}$ 的完工时间和为

$$n_i t_2 + \sum_{k=1}^{n_i} (n_i - k + 1) p_{ik} + m_1 (t_2 + p_i) + \sum_{k=1}^{m_1} (m_1 - k + 1) p_{hk} \quad (8)$$

比较式(2), (3), (7), (8) 可知, $\pi$ 比 $\pi_2$ 的总完工时间多

$$(n_i - n_j)(t_1 - t_2) + (m_1 - m_2)(t_1 - t_2) \quad (9)$$

由于 $n_i > n_j, t_1 > t_2, m_1 \geq m_2$, 因此式(9) 大于 0, 与 $\pi$ 是最优调度矛盾。( 证毕)

由于任意两个任务链之间可能不满足定理 1 的条件, 因此不能由此解得最优调度方案。但是对于满足定理 1 条件的任务链, 则可确定其加工顺序, 因此在利用枚举算法或分枝定界方法求解问题( 1) 时, 利用定理 1 的结论可减少计算量。

## 3　两种可解情况

由定理 1 的结论, 对问题( 1) 的两种特殊情况可得到最优调度方案。

**定理 2**　若每个链所含任务数均相同, 即 $n_1 = n_2 = \cdots = n_k$, 则将各个链按链加工时间不减排列即为最优调度方案。

**定理 3**　若每个链加工时间相等, 即 $p_1 = p_2 = \cdots = p_k$, 则将各个链按链所含任务数不增排列即为最优调度方案。

定理 2 和定理 3 的结论由定理 1 的证明过程即可得到。显然 $n_1 = n_2 = \cdots = n_k = 1$ 时, 定理 2 为 SPT 规则。下面给出两个数值例子。

**例 1**　$m = 2, \quad n = 9, \quad k = 3$

$L_1: T_1 \rightarrow T_2 \rightarrow T_3$

$L_2: T_4 \rightarrow T_5 \rightarrow T_6$

$L_3: T_7 \rightarrow T_8 \rightarrow T_9$

$p = (2, 3, 1, 2, 1, 2, 3, 2, 3)$

由定理 2 得最优调度方案: $L_2, L_1, L_3$, 总完工时间为 54。

**例 2**　$m = 2, \quad n = 9, \quad k = 3$

$L_1: T_1 \rightarrow T_2$

$L_2: T_3 \rightarrow T_4 \rightarrow T_5 \rightarrow T_6$

$L_3: T_7 \rightarrow T_8 \rightarrow T_9$

$p = (4, 1, 1, 2, 1, 1, 2, 2, 1)$

由定理 3 得最优调度方案: $L_2, L_3, L_1$, 总完工时间为 43。

## 4　结　　论

本文讨论任务的加工不允许中断, 任务间具有链约束平行机调度问题, 目标函数是极小化总完工时间。对于一般情况, 给出了最优调度满足的必要条件; 对于两种特殊情况, 给出了最优调度方案, 并利用具体例子对定理的应用做出解释。对于一般情况, 虽然不能用本文结论直接得到最优调度方案, 但在枚举算法和分枝定界算法利用本文的 3 个结论可使计算量减少, 较快得到最优调度方案。另外, 本文的全部结论可以推广处理机具有准备时间的情况[3]。证明完全类似, 不再赘述。

参考文献:

[1] M Pinedo. Scheduling-theory, algorithms and systems [M]. New Jersey: Prentice Hall, Englewood Cliffs, 1995.

[2] J Du, J Y T Leung, G H Young. Scheduling chain-structured tasks to minimize makespan and mean flow time[J]. Information and Computation, 1991, 92(3): 219-236.

[3] Lee C Y. Parallel machine scheduling with nonsimultanous machine available time[J]. Discrete Appl Math, 1991, 30(1): 53-61.