

## 等待时间受限的HFSP及其拉格朗日松弛算法

HFSP with limited waiting time and its lagrangian relaxation algorithm

丁小丽<sup>1,2</sup>, 朱军<sup>1</sup>, 刘昶<sup>1</sup>

DING Xiao-li<sup>1,2</sup>, ZHU Jun<sup>1</sup>, LIU Chang<sup>1</sup>

(1.中国科学院沈阳自动化研究所, 沈阳 110016; 2.中国科学院大学, 北京 100049)

**摘要:** 等待时间受限的混合流水车间调度问题 (HFSP) 广泛存在于钢铁生产, 玻璃加工和塑料等行业。目前, 关于等待时间受限的HFSP的研究较少。首先建立了等待时间受限的混合流水车间调度问题模型, 然后设计了基于工件分解策略的拉格朗日松弛算法来进行求解。该算法通过将机器容量约束松弛到目标函数中, 将得到的松弛问题进而分解为一系列易于求解的工件级子问题来进行求解。最后对设计的算法进行仿真验证, 测试结果表明所设计的拉格朗日松弛算法能够在较短的时间内产生较好的近优解。

**关键词:** 混合流水车间; 等待时间受限; 拉格朗日松弛算法

中图分类号: TP315

文献标识码: A

文章编号: 1009-0134(2015)07(上)-0067-04

Doi: 10.3969/j.issn.1009-0134.2015.07(上).18

### 0 引言

混合流水车间调度问题 (Hybrid Flow-shop Scheduling Problem, HFSP) 是由Salvador于1973年首先提出来的。HFSP是一类复杂的组合优化问题, 它是标准的flow-shop调度和并行机调度问题的综合, 相当普遍地存在于化工、钢铁、制药等流程工业中<sup>[1]</sup>。

等待时间受限的混合流水车间调度问题广泛存在于钢铁生产, 玻璃加工和塑料等行业, 它的形成是由于生产过程中的高温连续性, 要求工件在相邻两个阶段之间保持一定的温度, 例如在钢铁生产过程中的炼钢-连铸-热轧过程中, 连铸机对温度有严格的要求, 为避免温度降低影响钢水质量, 不允许钢水在工序间有过长的等待时间。因此, 可将两阶段之间的温度要求转换为对等待时间的要求, 从而形成了等待时间受限的混合流水车间调度问题。

目前, 关于等待时间受限的HFS调度问题的研究较少。文献[2]研究了具有零缓冲能力和有限等待时间的多处理器HFS问题, 以总加权拖期最小为目标, 基于离散时间和混合整数线性规划提出了一种精确求解算法。文献[3]研究了以最小化makspan为目标的等待时间受限的HFSP, 提出了一种启发式-禁忌算法来求解。文献[4]针对考虑交货期和等待时间受限的HFS调度问题, 提出了一种结合回溯和邻域搜索的混合算法来进行求解。文献[5]研究了以最小化生产等待时间为目标的带工件和工位等待时间的混合流水车间调度问题, 提出运用差分进化算法

来进行求解。文献[6]研究了以最小化各工件的提前/拖期惩罚和为目标的带交货期和释放期的等待时间受限的HFS调度问题, 采用约束满足优化算法和指派规则的混合算法来进行求解。

Gupta J N D<sup>[7]</sup>已经证明即使只有两个阶段且其中只有一个阶段包含多台并行机的HFSP是NP难问题, 因此, 本文要研究的更为复杂的HFSP也是典型的NP难问题。拉格朗日松弛算法<sup>[8,9]</sup>是解决生产调度问题的有效方法之一, 可用于NP难问题的求解, 它通过将复杂约束引入到目标函数, 可将原问题分解为多个简单的容易求解的子问题, 大大降低了问题的求解难度, 可以获得问题的近优解。目前对于等待时间受限的混合流水车间调度问题的求解在算法方面集中于精确求解算法或者智能优化类的近似算法。因此本文创新性的将拉格朗日松弛算法用于等待时间受限的混合流水车间调度问题研究中, 拓展了拉格朗日松弛算法的应用范围和混合流水车间<sup>[10,11]</sup>调度理论。

本文针对等待时间受限的混合流水车间调度问题, 首先建立了以最小化加权完成时间为目标的HFSP模型, 然后在此基础上设计相关的拉格朗日松弛算法来进行求解, 最后通过实验数据验证算法的可行性和有效性。

### 1 HFSP问题描述及建模

所研究的等待时间受限的HFSP可描述如下:  $n$  个工件在流水线上经过相同的 $s$ 个阶段的加工, 每个阶

收稿日期: 2015-03-16

基金项目: 863计划 (2013AA040705-1)

作者简介: 丁小丽 (1990-), 女, 河南人, 硕士研究生, 研究方向为排产与调度。

段  $j(j \in J, J = \{1, 2, \dots, s\})$  上有  $M_j$  台同构并行机, 每个工件  $i(i \in I, I = \{1, 2, \dots, n\})$  可以在阶段  $j$  的任意一台机器上进行加工, 已知工件  $i$  在阶段  $j$  的加工时间为  $p_{ij}$ , 同时工件  $i$  在相邻阶段  $j$  和  $j+1$  之间的等待时间不超过一定的时间上限  $a_j$ 。其中, 每台机器一次最多加工一个工件, 每一个工件在任意时刻最多只能在一台机器上进行加工, 且加工过程是连续的, 不允许发生中断。调度问题就是确定各个工件在各阶段的加工完成时间  $C_{ij}$ , 从而使得总加权完成时间达到最小。

为方便建模, 令  $\omega_i$  表示工件  $i$  的权重; 为一个无限长的调度计划期;  $\delta_{ijl}$  为一个 0-1 变量, 当时刻  $l(l \in T, T = \{1, 2, \dots, L\})$  时工件  $i$  在阶段  $j$  加工时为 1, 否则为 0;

根据以上描述, 所建立的等待时间受限的 HFSP 模型如下:

$$\min \sum_{i=1}^n C_{is} \quad (1)$$

s. t.

$$\sum_{i=1}^n \delta_{ijl} \leq M_j, \forall j, l \quad (2)$$

$$\sum_{j=1}^s \delta_{ijl} \leq 1, \forall i, l \quad (3)$$

$$C_{ij} \leq C_{i,j+1} - p_{i,j+1}, \forall i, j = 1, \dots, s-1 \quad (4)$$

$$C_{i,j+1} - p_{i,j+1} - C_{ij} \leq \alpha_j, \forall i, j = 1, \dots, s-1 \quad (5)$$

$$\sum_{l=1}^L \delta_{ijl} = p_{ij}, \forall i, j \quad (6)$$

$$l\delta_{ijl} \leq C_{ij}, \forall i, j, l \quad (7)$$

$$C_{ij} - p_{ij} + 1 \leq l + L(1 - \delta_{ijl}), \forall i, j, l \quad (8)$$

$$\delta_{ijl} \in \{0, 1\}, \forall i, j, l \quad (9)$$

$$C_{ij} \geq 0, \alpha_j \geq 0, \forall i, j \quad (10)$$

上述模型中目标函数(1)为所有工件的总加权完成时间。约束(2)表示机器容量约束, 即同一时刻在同一阶段进行加工的工件数不超过该阶段的机器数; 约束(3)表示同一工件在同一时刻最多只能在加工过程中的一个阶段进行加工; 约束(4)为工件的时间优先级约束, 表示工件只有在完成上一阶段的加工之后才能进入下一阶段的加工; 约束(5)表示工件在相邻阶段之间的等待时间不超过一定的上限; 约束(6)~(8)表示机器占用约束; 约束(9)和约束(10)为相关变量的取值范围定义。

等待时间受限的 HFSP 调度问题模型就是在满足约束(2)~(10)的条件下将工件安排到各阶段的机器上进行加工, 使得所有工件的总加权完成时间达到最小。

## 2 算法设计

拉格朗日松弛(LR)算法是解决生产调度问题的有效方法之一, 可用于 NP 难问题的求解, 它通过将复

杂约束引入到目标函数, 可将原问题分解为多个简单的容易求解的子问题, 大大降低了问题的求解难度, 可以获得问题的近优解。整个求解过程如下: 首先通过引入一组拉格朗日乘子将复杂约束松弛到目标函数中, 形成拉格朗日松弛问题; 然后将得到的 LR 问题分解为一系列简单的子问题进行求解, 最后对于松弛问题的解一般需要通过可行化来构造原问题的可行解; 重复上述过程直至达到停止准则。可行解的目标函数值提供了一个最优解的上界, 而对偶目标函数值为最优解的一个下界, 这样拉格朗日松弛算法同时也为衡量解的质量提供了一个标准, 解的质量可以通过两者之间的对偶间隙来进行衡量(对偶间隙 = [可行解的目标值 - 对偶目标值] / 对偶目标值)。

从上节所建立的模型中可以看出, 约束(3)~(5)都耦合了不同的阶段, 如果要基于阶段进行分解需要将这三个约束都进行松弛, 松弛的约束越多, 所得到的对偶目标值就越松, 而只有约束(2)耦合了不同的工件, 因此, 本文所设计的拉格朗日松弛算法采用基于工件进行分解的策略, 下面将详细的介绍拉格朗日松弛过程、子问题的求解、乘子的更新以及可行解的构造过程。

### 2.1 拉格朗日松弛

引入拉格朗日乘子向量 将约束(2)松弛到目标函数(1)中, 形成下面的拉格朗日松弛问题:

$$\begin{aligned} LR(\lambda) &= \min \left[ \sum_{i=1}^n \omega_i C_{is} + \sum_{j=1}^s \sum_{l=1}^L \lambda_{jl} (\sum_{i=1}^n \delta_{ijl} - M_j) \right] \\ &= - \sum_{j=1}^s \sum_{l=1}^L \lambda_{jl} M_j + \min \sum_{i=1}^n [\omega_i C_{is} + \sum_{j=1}^s \sum_{l=1}^L \lambda_{jl} \delta_{ijl}] \end{aligned} \quad (11)$$

满足约束(3)~(10)和:

$$\lambda_{jl} \geq 0, j \in J, l \in T \quad (12)$$

通过不断的逼近对偶问题的上界得到拉格朗日对偶问题:

$$LD(\lambda) = \max LR(\lambda) \quad (13)$$

满足约束(3)~(10)和(12)。

当乘子给定时, 式(11)中的前半部分是一个常数, 将后半部分基于工件进行分解, 得到多个工件级子问题, 工件  $i$  的工件级子问题可表示为:

$$LR_i(\lambda) = \omega_i C_{is} + \sum_{l=1}^L \lambda_{jl} \delta_{ijl}, j = s \quad (14)$$

满足约束(3)~(10)和(12)。整理后得到:

$$LR_i(\lambda) = \omega_i C_{is} + \sum_{l=1}^L \lambda_{jl} \delta_{ijl}, j = s \quad (15)$$

$$LR_i(\lambda) = \sum_{l=1}^L \lambda_{jl} \delta_{ijl}, j = 1, 2, \dots, s-1 \quad (16)$$

因此,  $LR(\lambda)$  可归结为:

$$LR(\lambda) = -\sum_{j=1}^s \sum_{l=1}^L \lambda_{jl} M_j + \sum_{i=1}^n \min LR_i(\lambda) \quad (17)$$

满足式(3)~(10)和(12)。

## 2.2 子问题的求解

利用反向动态规划来求解工件级子问题。当给定工件*i*和一组乘子时，利用以下递推公式来计算出工件*i*在各个阶段的累积函数值：

$$F(j, C_{ij}) = \begin{cases} LR_i(\lambda), & j = s \\ LR_i(\lambda) + \min[F(j+1, C_{i,j+1})], & j < s \end{cases} \quad (18)$$

## 2.3 构造可行解

由于松弛了机器容量约束(2)，得到的松弛问题的解往往是不可行的，需要进行可行化，构造可行解的过程如下：首先根据得到的工件的完成时间计算出工件在各个阶段的开始加工时间表。在每个加工阶段，按照工件在该阶段的开始加工时间的增序进行排列，依次将工件安排在可利用的机器上；然后检查各个工件相邻阶段之间的等待时间是否满足要求，如果某工件在相邻两阶段之间的等待时间超过了给定的时间上限，则将该工件在前一阶段的开始加工时间后移，与后一个工件进行两两交换，直到前后两个阶段之间的等待时间满足要求为止。

## 2.4 更新拉格朗日乘子

采用次梯度算法来对拉格朗日乘子进行更新。主要步骤如下：

STEP1：迭代次数  $n=1$  时，初始化拉格朗日乘子及相关参数。

STEP2：求解对偶问题和次梯度  $g(\lambda^n)$ ，其中  $\lambda^n$  是第  $n$  次迭代过程中的拉格朗日乘子向量  $\{\lambda_{ij}\}$ 。

STEP3：更新乘子。根据下列式子更新拉格朗日乘子：

$$\lambda^{n+1} = \lambda^n + \pi^n g(\lambda^n) \quad (19)$$

其中  $\pi^n$  是  $n$  次迭代步长， $\pi^n$  通过下式计算得到：

$$\pi^n = \varepsilon \frac{G^U - G^n}{\|g(\lambda^n)\|}, 0 < \varepsilon \leq 1 \quad (20)$$

上式中  $G^U$  是当前得到的最优的可行目标值的上界； $G^n$  是  $n$  次迭代的  $LR(\lambda)$  值； $\varepsilon$  初始值设为 1， $G^n$  上升时保持不变，当  $G^n$  在若干次迭代中未发生变化时则取其一半。

STEP4：判断是否满足停止条件。若对偶间隙小于一个很小的数或达到了最大迭代次数，程序停止；否则返回STEP2进行下一次迭代。

图1为拉格朗日松弛算法流程图。

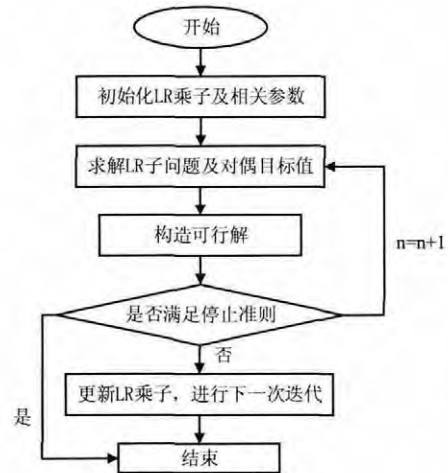


图1 拉格朗日松弛算法流程图

## 3 仿真结果及分析

为了对所提出的拉格朗日松弛算法进行可行性和有效性的验证，采用MATLAB对上述拉格朗日松弛算法进行编程，并在Intel(R) Core(TM) CPU 3.2GHz的计算机上运行。

实验中相关的参数设置如下：待加工工件数  $n \in \{10, 20, 40\}$ ；加工阶段数  $s \in \{3, 4, 5\}$ ；每个阶段上的并行机器数  $M_j \in \{2, 3, 4\}$ ；各工件的权重服从  $[1, 5]$  上的均匀分布；各工件在各阶段的加工时间  $p_{ij}$  服从区间  $[1, 10]$  的均匀分布；工件在相邻阶段间的等待时间上限  $\alpha_j \in [5, 10]$ ；且实验中采用最大迭代次数300作为停止条件。

上述  $\{n, s, M_j\}$  的组合共产生27种不同规模的问题，在实验中对于每种规模问题随机生成5个实例，取他们的平均值作为该规模问题的运行结果。算法的性能采用对偶间隙GAP ( $GAP = (G^U - G^n) / G^n \times 100\%$ ) 和运行时间TIME(秒)这两个指标来进行衡量。表1为不同规模问题的仿真结果。

分析表1~表3中的数据可以看出：

1) 对于同一问题规模（当工件数和阶段数一定时），随着并行机器数的增加，可利用的机器资源增加，运行时间有所减少，对偶间隙也得到了改善；

2) 对于同一规模问题，随着相邻阶段之间的等待时间的增大，算法的运行时间相应加长，得到的解质量略有下降（对偶间隙略有增加）。即和对解的质量（对偶间隙）的影响很小，由此说明所设计的拉格朗日松弛算法对于不同的等待时间均能得到较好的近优解。

3) 对于不同规模问题而言，时算法的运行时间随之增加。且当由5增加到10时，运行时间有较大幅度的增加。

图2为不同工件数在加工阶段数为5，各个阶段的机器数为4，等待时间为10的情况下对偶间隙随迭代次数

变化趋势图，三条曲线分别反应了工件数为10，20和40时的对偶间隙变化，从图中可以看出：

1) 所设计的拉格朗日松弛算法在迭代前期有较好的收敛速度，随着迭代次数的增加，收敛速度逐渐减缓并最终趋向于收敛。

2) 在加工阶段和机器数量相同时，在同一迭代次数下，规模较大的问题的对偶间隙比规模较小的问题的对偶间隙要相对大一些，即规模较小的问题所得到的解的质量相对好一些。

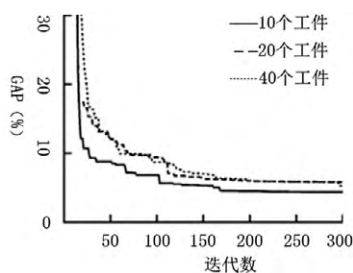


图2 不同工件数时对偶间隙随迭代次数变化趋势

综合上述分析可以看出，本文所设计的拉格朗日松弛算法能够在较短的时间内产生较好的近优解，可以用于等待时间受限的混合流水车间调度问题中。

## 4 结束语

本文针对由于生产过程中的高温连续性等要求而产生的等待时间受限的混合流水车间问题进行问题建模和算法的研究。首先建立了等待时间受限的混合流水车间调度问题模型，然后在此基础上设计相应的拉格朗日松弛算法来进行求解。该算法通过将机器容量约束松弛到目标函数中，进而将得到的松弛问题分解为一系列易于求解的工件级子问题来进行求解，并利用动态规划算法来求解子问题，然后对得到的松弛问题的解利用一种启发式算法来进行可行化，并通过次梯度算法来不断更新乘子。最后对设计的算法进行仿真验证，测试结果表明所设计的拉格朗日松弛算法能够在较短的时间内产生较好的近优解。

## 参考文献：

- [1] 王凌,周刚,许焯,等.混合流水线调度研究进展[J].化工自动化及仪表,2011,38(1):1-8.
- [2] Gicquel C, Hege L, Minoux M, et al. A discrete time exact solution approach for a complex hybrid flow-shop scheduling problem with limited-wait constraints[J]. Computers & Operations Research, 2012, 39(3): 629-636.
- [3] Liu S Y, Cui J S, Li Y. Heuristic-Tabu algorithm for hybrid flow-shop scheduling with limited waiting time[A]. Wuhan: Proceeding of the International Symposium on Computational Intelligence and Design[C]. 2008: 233-237.
- [4] 尹兆涛,李铁克.考虑交货期和等待时间受限的HFS调度问题的混合算法[J].工业工程,2009,12(1):79-83.
- [5] 王长涛,刘春光,胡平东,等.混合流水车间等待时间优化研究[J].沈阳建筑大学学报(自然科学版),2012,28(2):368-374.
- [6] 肖拥军,李铁克,尹兆涛.考虑特殊时间约束的混合流水车间调度[J].计算机工程与应用,2010,46(8):205-207,231.
- [7] Gupta J N D. Two stage, hybrid flow-shop scheduling problem[J]. Journal of Operational Research Society, 1988, 39(1): 359-364.
- [8] 轩华,唐立新.实时无等待HFS调度的一种拉格朗日松弛算法[J].控制与决策,2006,21(4):376-380.
- [9] 杜书魁.需准备时间的FFS调度的一种拉格朗日松弛算法[J].科学技术与工程,2012,12(6):1272-1277.
- [10] 张其亮,陈永生.带有阻塞限制的混合流水车间调度问题的混合粒子群求解算法[J].信息与控制,2013,42(2):252-257.
- [11] 屈国强.瓶颈指向的启发式算法求解混合流水车间调度问题[J].信息与控制,2012,41(4):514-521,528.

表1 n=10的仿真结果

	s × M <sub>i</sub>	WT=5		WT=10	
		TIME	GAP	TIME	GAP
n=10	3 × 2	0.32	2.78	2.86	3.18
	3 × 3	0.31	2.03	2.50	2.19
	3 × 4	0.27	1.75	1.92	1.70
	4 × 2	0.40	1.80	2.93	2.16
	4 × 3	0.31	3.29	2.84	1.79
	4 × 4	0.34	2.30	1.56	3.07
	5 × 2	0.97	1.98	4.89	1.29
	5 × 3	0.72	2.73	3.27	2.03
Avg		0.50	2.18	2.97	2.25

表2 n=20的仿真结果

	s × M <sub>i</sub>	WT=5		WT=10	
		TIME	GAP	TIME	GAP
n=20	3 × 2	2.37	2.48	5.73	3.56
	3 × 3	2.35	2.33	5.39	2.99
	3 × 4	2.27	1.79	4.92	2.73
	4 × 2	2.93	2.80	7.93	2.16
	4 × 3	2.43	3.20	6.85	1.79
	4 × 4	2.79	2.93	6.16	1.07
	5 × 2	2.97	1.48	9.29	2.29
	5 × 3	2.42	2.79	8.27	2.03
Avg		2.60	2.28	6.86	2.31

表3 n=40的仿真结果

	s × M <sub>i</sub>	WT=5		WT=10	
		TIME	GAP	TIME	GAP
n=40	3 × 2	18.78	6.20	29.82	6.32
	3 × 3	16.43	4.29	27.33	5.10
	3 × 4	16.20	3.88	25.90	3.29
	4 × 2	20.34	5.90	31.29	6.19
	4 × 3	19.20	5.12	29.80	4.20
	4 × 4	17.89	4.03	27.39	5.27
	5 × 2	25.39	4.97	34.20	5.22
	5 × 3	23.92	3.84	29.38	3.28
Avg		19.78	4.71	29.27	4.72