可拆分平行机排序问题的一个启发式算法

郑秋亚[1,2]， 刘三阳[1]， 杨尊袍[3]

(1. 西安电子科技大学 理学院，陕西 西安 710071；2. 长安大学 理学院，陕西 西安 710064；3. 空军工程大学 理学院，陕西 西安 710051)

摘 要：为缩短工件的完工时间，将极小化最大完工时间的平行机排序问题作为研究目标。在此问题中，允许同一工件拆分成多个子工件在不同的机器上同时加工，同一工件的任何 2 个子工件不可在同一台机器上加工。与以往研究不同，对工件的拆分方式进行了限制，即工件拆分后所得子工件的长度不能小于给定的阀值，且工件拆分次数尽量少，这是一个 NP 难问题。借助于 LPT 算法的思想，提出了一个求解该问题的启发式算法，实现了工件的自动拆分和工件到机器上的自动分配。通过多个实例对文中算法进行了测试，数值结果表明：该算法可行、稳定性良好，适用于工件拆分方式具有类似限制的平行机排序问题的方案决策。

关键词：启发式算法；最大完工时间；排序；拆分；平行机

DOI：10. 3969/j. issn. 1009 – 3516. 2010. 04. 017

中图分类号：TP301 **文献标识码**：A **文章编号**：1009 – 3516(2010)04 – 0084 – 05

极小化最大完工时间的 n 个独立工件在 m 台同型号平行机上的经典排序问题是 NP 难的，即使只有 2 台机器($P_2//C_{\max}$)它也是 NP 难的[1]。突破经典排序的第 1 个基本假设，允许单个工件拆分成多个子工件在不同的机器上同时加工，这类问题被称为可拆分平行机排序问题[2-3]。

有关可拆分平行机排序方面的文献并不是很多，Xing 和 Zhang[4-5] 研究了可拆分平行机排序问题在各种费用目标下的算法及复杂性，指出：当工件任意连续可分时 $P_m//C_{\max}$ 是多项式时间可解问题，而工件带安装时间的同类问题为 NP 难的[6-9]。迄今为止，有关可拆分平行机排序的研究多数是在工件任意连续可分的条件下展开的，而实际问题中，经常出现的是允许工件拆分但对工件拆分方式提出某些限制。例如：多处理计算机系统中基于区域分解方法的并行计算中的网格数据划分和任务分配问题，当网格块数少于处理机的个数或网格块大小参差不齐时，想要尽快的完成计算任务，通常采用的解决办法是对网格(工件)进行拆分，但要求网格拆分后所得子网格的规模(工件长度)不能太小，否则，计算无法实现；同时应使网格拆分的次数尽量少，以保证整体计算的收敛性和稳定性。这种工件拆分方式有限制的排序问题往往是 NP 难的。本文从实际问题出发，在允许工件拆分的假设下，对工件拆分方式提出限制，研究极小化最大完工时间的平行机排序问题，给出求解此类问题的一个近似算法，并通过实例说明该算法的性能。

1 问题

考虑 n 个独立工件在 m 个同型机上的排序问题，目标是极小化最大完工时间，允许单个工件拆分成多个子工件在不同的机器上同时加工，对工件拆分方式的限制是：工件拆分后所得子工件的长度不能小于给定的阀值，且工件拆分次数尽量少。

给定 m 台同型机 $M_1, M_2, \cdots, M_m$，n 个相互独立的工件 $J_1, J_2, \cdots, J_n$，工件 J_j 的加工时间(长度)$p_j > p_{\min}$，

* **收稿日期**：2009 – 11 – 30
基金项目：国家自然科学基金资助项目(60574075)
作者简介：郑秋亚(1964 –)女，陕西乾县人，高级工程师，博士生，主要从事最优化方面研究；
E – mail：qiuyazheng@126. com
刘三阳(1959 –)男，陕西西安人，教授，博士生导师，博士(后)，主要从事最优化和控制方面研究.

$p_{\min}$为给定工件长度的最小阀值。用 $Q_{j,k}$表示工件 J_j在机器 M_k 上的加工时间,$Q_{j,k}=0$ 表示机器 M_k 未加工工件 J_j,$Q_{j,k}>p_{\min}$表示机器 M_k 对工件 J_j 有加工,那么,所有机器加工工件 J_j 的时间之和应为 p_j,即:

$$\sum_{k=1}^{m} Q_{j,k} = p_j \quad , \quad j = 1,2,\cdots,n \tag{1}$$

用 C_j 表示工件 J_j 的完工时间,D_i 表示机器 M_i 完成其上所有工件加工所用时间,则:

$$D_i = \sum_{j \in W_i} Q_{j,i} \quad , \quad i = 1,2,\cdots,m \tag{2}$$

式中 W_i 表示在机器 M_i 上加工的工件组成的集合。

令 $C_{\max} = \max\limits_{1\leqslant j\leqslant n} C_j$ 或 $C_{\max} = \max\limits_{1\leqslant i\leqslant m} D_i$,则目标为:

$$C^*_{\max} = \min C_{\max} \tag{3}$$

m 台机器加工 n 个工件的平均完工时间为 $\sum\limits_{j=1}^{n} p_j/m$,由于 $\max\limits_{1\leqslant i\leqslant m} D_i \geqslant \sum\limits_{j=1}^{n} p_j/m$,所以近似最优解的下界为:

$$C_L = t_{\text{ave}} = \sum_{j=1}^{n} p_j/m \tag{4}$$

本文称长度大于 m 台机器上的平均长度 $\sum\limits_{j=1}^{n} p_j/m$ 与 $p_{\min}$ 之和的工件为超长工件。

实际应用中,为了使工件拆分次数最少,只拆分超长工件,其方法分为 2 步:首先,拆分超长工件为 2 个子工件,使其中一个子工件的加工时间为机器的平均完工时间,并将这个子工件分配给一台机器;其次,将所有非超长工件和所有超长工件的剩余部分在未指派机器上进行排序。上述第 2 步为一经典平行机排序问题,当没有工件被拆分时,本文问题就是一个经典平行机排序问题,所以, 在工件拆分次数最少,且拆分后子工件的长度不小于给定阀值限定条件下的平行机排序问题是 NP 难的。

2　启发式算法

LPT(Longest Processing Time First)算法是 Graham 提出的平行机排序问题的著名的启发式算法,其复杂度为 $O(n\log n)$,因其只考虑“眼前”利益,结构简单,所以成为几乎每个 NP 难排序问题首先被提出并受到研究的算法,本文对 LPT 算法加以改进,将其推广应用到拆分方式有限制的平行机排序问题。不失一般性,令 $p_{\min}=t_{\text{ave}}\text{esp}$,则可求出 $\text{esp}=p_{\min}/t_{\text{ave}}$。

2.1　常规算法

步骤 1　拆分超长工件;

步骤 2　用 LPT 算法进行排序。

2.2　近似算法

步骤 1　求 m 台机器完成所有工件加工需要的平均时间 $t_{\text{ave}}=\sum\limits_{j=1}^{n}p_j/m$;

步骤 2　将未被分配的工件按加工时间不增排序建立链表 1;

步骤 3　当链表 1 非空时:①从链表 1 中依次取工件 J_j,将其加载到当前速度最快的机器 M_i 上,求出机器 M_i 当前完工时间 D_i;②若 $D_i>t_{\text{ave}}(1+\text{esp})$,将工件 J_j 按图 1 所示拆分成 2 个子工件, 将子工件 J_j^{remain} 插入到链表 1 中;将工件 J_j(此处为子工件)分配给机器 M_i,直到链表 1 为空。

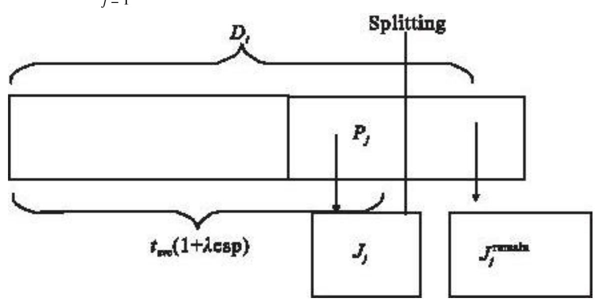

图 1　工件拆分示意图(0 < λ ≤1)

Fig. 1　Splitting(0 < λ ≤1)

3　实例模拟及结果分析

以下叙述中,时间单位均为 s;取 $p_{\min}=0.03t_{\text{ave}}$,则 $\text{esp}=0.03$。

实例 1:11 个可拆分的工件,其加工时间见表 1。

表 1 工件加工时间

Tab. 1 Processing times

工件	1	2	3	4	5	6	7	8	9	10	11
加工时间	40. 16	37. 61	33. 39	32. 13	142. 80	26. 62	26. 69	13. 87	8. 57	29. 48	12. 90

3.1 与常规方法的比较

方案 1:由 6 台机器加工。

6 台机器加工 11 个工件的平均时间即最优解下界为 67. 37,工件长度的最小阀值 $p_{\min}=2.02$。5 号工件的加工时间为 142. 8,大于工件的平均完工时间与工件长度的最小阀值之和 69. 39,所以被拆分成 3 个长度分别为:67. 57、67. 57 和 7. 65 的子工件,其中长度为 67. 57 的 2 个子工件分别被分配给 1 号和 2 号机器加工,而长度为 7. 65 的子工件和另外 10 个工件在 3 号到 6 号机器上进行排序。

表 2 是 5 号工件被拆分 2 次后,长度为 7. 65 的子工件和另外 10 个工件在 3 号到 6 号机器上的最优排序,对应的最优解为 67. 66。

表 3 是 6 台机器加工 11 个工件本文算法的近似结果,其中包括:工件的加工方案、对应于此加工方案的工件 J_j 在机器 M_k 上的加工量 $Q_{j,k}$ 以及工件 J_j 的完工时间和机器 M_k 的完工时间。由表 3 可看出:5 号工件被拆分成 3 个长度分别为:67. 57、67. 57 和 7. 65 的子工件,其中长度为 67. 57 的 2 个子工件分别被分配给 1 号和 2 号机器加工,而长度为 7. 65 的子工件和另外 10 个工件在 3 号到 6 号机器上进行排序,由于在 3 号到 6 号机器上排序期间再没有工件被拆分,所以此排序实际上是按 LPT 算法进行的。也就是说,在此例中本文算法和常规方法结果相同,所得近似解为 69. 26,相对于最优解下界的误差为 2. 8%。

表 2 最优解

Tab. 2 Optimum solution

机器	工件序列	工件加工量	工件完工时间	机器完工时间	近似解
1	5	67. 57	67. 57	67. 57	
2	$\boxed{5}$	67. 57	67. 57	67. 57	
3	1 – 8 – 11	40. 16 – 13. 87 – 12. 90	40. 16 – 54. 03 – 66. 93	66. 93	
4	2 – 10	37. 61 – 29. 48	37. 61 – 67. 09	67. 09	
5	3 – 6 – $\boxed{5}$	33. 39 – 26. 62 – 7. 65	33. 39 – 60. 01 – 67. 66	67. 66	67. 66
6	4 – 7 – 9	32. 13 – –26. 69 – 8. 57	32. 13 – 58. 82 – 67. 39	67. 39	

表 3 近似最优解

Tab. 3 The approximate optimum solution

机器	工件序列	工件加工量	工件完工时间	机器完工时间	近似解
1	5	67. 57	67. 57	67. 57	
2	$\boxed{5}$	67. 57	67. 57	67. 57	
3	1 – 8 – 11	40. 16 – 13. 87 – 12. 90	40. 16 – 54. 03 – 66. 93	66. 93	
4	2 – 6	37. 61 – 26. 62	37. 61 – 64. 23	64. 23	
5	3 – 7 – 9	33. 39 – 26. 69 – 8. 57	33. 39 – 60. 08 – 68. 65	68. 65	
6	4 – 10 – $\boxed{5}$	32. 13 – 29. 48 – 7. 65	32. 1 – 61. 61 – 69. 26	69. 26	69. 26

方案 2:由 4 台机器加工。

表 4 是 4 台机器加工 11 个工件的近似方案,其最优解下界为 101. 06,工件长度的最小阀值 $p_{\min}=3.03$。如果用常规方法,首先,5 号工件被拆分成 2 个子工件;其次,用 LPT 方法将剩余工件在 2 号、3 号和 4 号机器上进行排序,得近似解为 106. 19,相对于最优解下界的误差为 5. 08%;用本文算法,在工件分配的过程中,除 5 号工件外,当 9 号工件被分配给 4 号机器时,4 号机器的完工时间为 106. 19,大于机器的平均完工时间 101. 06 与工件长度最小阀值 3. 03 之和,此时,9 号工件相对于 4 号机器当前的情形为超长工件,被拆分成 2 个子工件,分别由 2 和 4 号机器加工,最终本文近似解为 102. 62,相对于最优解下界的误差为 1. 54%。比传统方法近似解的误差降低了 3. 54%,说明本文算法确实起到了缩短工件完工时间的功效。

表 4　近似最优解

Tab. 4　The approximate optimum solution

机器	工件序列	工件加工量	工件完工时间	机器完工时间	近似解
1	5	101.36 101.36 101.36			
2	$\boxed{5}$ -10 -7 - $\boxed{9}$	41.44 -29.48 -26.69 -3.57	41.44 -70.92 -97.61 -101.18	101.18	
3	1 -4 -8 -11	40.16 -32.13 -13.87 -12.90	40.16 -72.29 -86.16 -99.06	99.06	
4	2 -3 -6 -9	37.61 -33.39 -26.62 -5.00	37.61 -71.00 -97.62 -102.62	102.62	102.62

3.2　算法性能测评

为考察本文算法的性能,引入负载均方差和负载相对均方差分指标[10]。负载均方差和负载相对均方差是评价各台机器上负载平衡性的性能指标:其值越小,说明各台机器上的负载越平衡,机器的完工时间就越短,算法的性能就越好。相对误差指近似解相对于最优解下界的误差。

3.2.1　不同机器加工相同工件

表 5 给出 11 个工件在不同机器上加工所得近似解以及算法的性能指标。由表 5 可以看出,本文近似解的相对误差在 0.32% -2.81% 之间,均小于 3% ,说明本文算法可信。相对误差并不随机器个数的增减而有较大幅度的变化,说明本文算法相对于机器台数性能稳定。性能指标说明每台机器上的负载相当平衡,说明本文算法整体性能良好。

表 5　计算结果和性能

Tab. 5　Computational results and performance

机器数	最优解下界	近似解	相对误差(%)	负载均方差	负载相对均方差(%)
3	134.72	135.15	0.32	0.162 6	0.120 7
7	57.75	58.81	1.84	0.531 0	0.919 5
12	33.69	33.97	0.83	0.082 8	0.245 7
24	16.84	17.21	2.20	0.207 3	1.231 0

3.2.2　同一机器加工不同工件

实例 2:17 个可拆分的工件,其加工时间见表 6,分别由 6 台、9 台和 12 台机器加工。

表 7 给出由 6 台、9 台和 12 台机器加工表 1 和表 6 中的工件所得的近似结果。可以发现:表 7 所示近似解的相对误差同样不大于 2.81% ,说明本文算法相对于工件规模性能稳定。

两方面数值实验的结果表明,本文算法性能良好,能给出好的近似最优解。

表 6　工件加工时间

Tab. 6　Processing times

工件	加工时间	工件	加工时间	工件	加工时间	工件	加工时间
1	191.64	2	71.81	3	32.64	4	14.69
5	14.69	6	5.70	7	14.99	8	14.99
9	5.71	10	5.26	11	105.26	12	138.72
13	129.25	14	12.57	15	77.11	16	47.86

表 7　计算结果

Tab. 7　Computational results

机器数	工件数	最优解下界	近似解	近似解/最优解下界	相对误差(%)
6	11	67.37	69.26	1.028 1	2.81
	16	167.15	168.40	1.007 5	0.75
9	11	44.91	45.87	1.020 9	2.09
	16	111.43	113.32	1.017 0	1.70
12	11	33.69	33.97	1.008 3	0.83
	16	83.58	85.60	1.024 2	2.42

4　结束语

对工件拆分方式有限制的极小化最大完工时间的平行机排序问题进行了研究,该问题是 NP 难的。借

用 LPT 算法的思想,提出了一个求解该问题的启发式算法,该算法实施在多个实例上所得近似最优解相对到最优解下界的误差不大于 2.81% 。该算法与常规方法相比,缩短了工件的完工时间。数值结果和各种性能指标说明了本文算法稳定,能给出好的决策方案。

参考文献:

[1] Karp R M. Reducibility Among Combinatorial Problems[M]//Miller R E, Thatcher J W. Complexity of Computer Computations. New York: Plenum Press, 1972: 85 - 103.

[2] Serafini P. Scheduling Jobs on Several Machines with the Job Splitting Property [J]. Operations Research, 1996, 44: 617 - 628.

[3] Lam K, Xing W. New Trends in Parallel Machine Scheduling [J]. International Journal of Operations & Production Management, 1997, 17(3): 326 - 338.

[4] 邢文训,张家伟. 可拆分平行机排序问题[J]. 运筹学学报,1998, 17(3):30 - 41.
XING Wenxun, ZHANG Jiawei. Splitting Parallel Machine Scheduling [J]. Or Transactions, 1998, 17(3): 30 - 41. (in Chinese)

[5] Xing W, Zhang J. Parallel Machine Scheduling with Splitting Jobs [J]. Discrete Applied Mathematics, 2000, 103: 259 - 269.

[6] Yalaoui F, Chu C. An Efficient Heuristic Approach for Parallel Machine Scheduling with Job Splitting and Sequence - dependent Setup Times [J]. IIE Transactions, 2003, 35(2): 183 - 190.

[7] Tahar D N, Yalaoui F, Chu C, et al. A Linear Programming Approach for Identical Parallel Machine Scheduling with Job Splitting and Sequence - dependent Setup Times [J]. International Journal of Production Economics, 2006, 99: 63 - 73.

[8] Shim S O, Kim Y D. A Branch and Bound Algorithm for An Identical Parallel Machine Scheduling Problem with A Job Splitting Property [J]. Computers & Operations Research, 2008, 35: 863 - 875.

[9] KIM Y D, SHIM S O, KIM S B, et al. Parallel Machine Scheduling Considering A Job Splitting Property[J]. International Journal of Production Research, 2004, 42: 4531 - 4546.

[10] 向建军,白欣,左继章. 一种用于实时集群的多任务负载均衡算法[J]. 计算机工程,2003,29(12):36 - 38.
XIANG Jianjun, BAI Xin, ZUO Jizhang. A Multitask Load Balancing Algorithm Used in Real - time Cluster System [J]. Computer Engineering, 2003, 29(12): 36 - 38. (in Chinese)

(编辑:徐楠楠)

A Heuristic Algorithm for Parallel Machine Scheduling Problem with Job Splitting Property

ZHENG Qiu - ya[1,2], LIU San - yang[1], YANG Zun - pao[3]

(1. Science School of Xidian University, Xi′an 710071, China; 2. Science School of Chang′an University, Xi′an 710064, China; 3. Science Institute, Air Force Engineering University, Xi′an 710051, China)

Abstract: An identical parallel machine scheduling problem with job splitting to minimize makespan is studied to decrease the completion time of the job. In this problem, each job can be split into sections, which can be processed in parallel on different machines. There is at most one part of each job on a machine. Different from researches in the scheduling literatures, there is a restriction for the split, i. e. the size of each split section cannot be smaller than a given value and the splitting actions should be kept as less as possible in the number of times. For this NP - hard problem, a heuristic algorithm is developed based on the LPT algorithm. Using the algorithm, job is split and assigned automatically. The performance of the algorithm is evaluated through a number of numerical instances. The results show that the algorithm is feasible and fine in stability. This algorithm can be applied to solving parallel machine scheduling problem, in which the split has the similar restriction.

Key words: heuristic algorithm; makespan; scheduling; splitting; parallel machine