Advances in Applied Mathematics 应用数学进展, 2023, 12(12), 5186-5192
Published Online December 2023 in Hans. https://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2023.1212509

Hans汉斯

带约定时间的多维任务负载平衡问题

吴建丽，代兵飞*

楚雄师范学院数学与计算机科学学院，云南 楚雄

收稿日期：2023年11月25日；录用日期：2023年12月19日；发布日期：2023年12月26日

摘　要

本文研究了两台平行机上带约定时间的多维任务负载平衡问题，该问题是传统负载平衡问题的推广，目标是最大化机器的提前工作量。带有约定时间的多维任务负载平衡问题涉及到在多个维度上平衡任务的分配，同时满足任务的约定时间。这种问题在许多实际应用中都很常见，比如生产调度、交通规划、资源分配等领域。对于该问题，当任务的维数为l时，本文设计了一个在线算法，算法的竞争比为$\left(\sqrt{5}-1\right)l$，然后设计了一个动态规划算法以得到问题的最优解，最后在动态规划的基础上采用舍入取整技术设计了一个FPTAS。

关键词

多维任务，在线算法，负载平衡，约定时间

Multidimensional Task Load Balancing Problem with a Common Due Date

Jianli Wu, Bingfei Dai*

School of Mathematics and Computer Science, Chuxiong Normal University, Chuxiong Yunnan

Received: Nov. 25th, 2023; accepted: Dec. 19th, 2023; published: Dec. 26th, 2023

Abstract

In this paper, we study multidimensional task loading problem on two parallel machines with a common due date. This problem is a generalization of the traditional load balancing problem, with the goal of maximizing the advance workload of machines. The multi-dimensional task load balancing problem with a common due date involves balancing the distribution of tasks in multiple dimensions while satisfying the due date time of tasks. This kind of problem is common in many

*通讯作者。

文章引用: 吴建丽, 代兵飞. 带约定时间的多维任务负载平衡问题[J]. 应用数学进展, 2023, 12(12): 5186-5192.
DOI: 10.12677/aam.2023.1212509

practical applications, such as production scheduling, transportation planning, resource allocation. For this problem, when the dimension of the task is l, we first designs an online algorithm with a competitive ratio of $\left(\sqrt{5}-1\right)l$, and the dynamic programming algorithm is designed to obtain the optimal solution of the problem. Finally, on the basis of dynamic programming, the FPTAS is designed by rounding technique.

Keywords

Multidimensional Task, Online Algorithm, Load Balancing, Due Date

1. 引言

带约定时间的负载平衡问题是一类具有广泛的实际问题。该问题常常发生在生产贸易中，客户与工厂约定一个交货时间 d，在 d 之前交付的商品，客户支付较高的价格，在 d 之后交付的商品，客户支付较低的价格。工厂为了最大化利润，需要制定生产计划，以使得在约定时间之前交付更多的商品。

对于带约定时间的两台平行机负载平衡问题，Chen 等人在文[1]中设计了一个竞争比为 $\sqrt{5}-1$ 的最优在线算法。当所有工件总的加工时间已知时，Chen 等人在文[2]中设计了一个竞争比为 6/5 的最优半在线算法。对于带约定时间和等级的两台平行机负载平衡问题，Xiao 等人在文[3]中设计了三个半在线算法，当已知低等级或高等级工件的加工时间之和时，分别设计了一个竞争比为 $\sqrt{5}-1$ 的最优半在线算法，当已知低等级与高等级工件的加工时间之和时，设计了一个竞争比为 6/5 的最优半在线算法。

向量负载平衡问题是把 n 个 l-维向量分配给 m 台机器加工以使得所有机器、所有维数的最大负载尽可能小。Graham 等人在文[4]中创造性的工作，使得单维负载均衡问题被广泛的推广。Chekuri 等人在文[5]首次提出离线多维向量负载平衡问题。当向量维数是任意常数时，Chekuri 等人在文[5]设计了一个 $O\left(\ln^2 l\right)$-近似算法。Meyerson 等人在文[6]中设计了一个 $O\left(\log l\right)$-近似算法。Im 等人在文[7]设计了一个 $O\left(\log l/\log\log l\right)$-近似算法。对于向量负载平衡问题，当向量维数 l 是固定常数时，Chekuri 在文[5]首次给出多项式时间近似方案，算法的运行时间是 $n^{g(\varepsilon,l)}$。Bansal 等人在文[8]设计了一个运行时间为 $\exp\left(\left(1/\varepsilon\right)^{O(l\log\log l)}\right)+nl$ 的 $\left(1+\varepsilon\right)$-近似算法。

Li 等人在文[9]中第一次提出带惩罚费用的向量负载平衡问题，并证明了该问题是 NP-hard。他们在多项式时间内设计了两个近似算法。Dai 等人在文[10]中提出了两台平行机上带惩罚费用的多维任务负载平衡问题，设计了一个 3-近似算法；在任务的维数为固定常数的情况下，在任务的维数 l 为固定常数的情况下，利用动态规划算法得到一个完全多项式时间近似方案；根据随机舍入方法，设计了一个近似算法，算法的近似比为 2.54；在多维任务的信息未知时，设计了一个在线算法，在线算法的竞争比为 $2.618l$。

在实际问题中，一项任务可能需要多种资源来完成，因此需要用多维向量描述任务的需求。在本文中，每个多维任务都有一个共同的约定时间，算法需要把每个多维任务调度到一台机器上加工。基于此，本文研究两台平行机上带约定时间的多维任务提前工作量最大化负载平衡问题，该问题记作 $P_2\left|d_j=d,VC\right|\max\left(X\right)$。

2. 问题描述及符号说明

两台平行机上带约定时间的多维任务负载平衡问题可以描述为：给定一个实例 $I=(M,J,p,d)$，两台平行机构成的集合 $M=\{M_1,M_2\}$，n 项任务构成的集合 $J=\{J_1,J_2,\cdots,J_n\}$，一个共同的约定时间 d。$p_j=(p_{j1},p_{j2},\cdots,p_{jl})^{\mathrm{T}}$ 是对应于任务 $J_j(j=1,\cdots,n)$ 的一个 l-维向量。任务 J_j 到达后，算法立刻分配给一台机器加工，$S_i(i=1,2)$ 表示分配给机器 M_i 的任务集。

$L_{i\max}$ 表示机器 M_i 负载向量的最大分量，

$$L_{i\max}=\max_k\sum_{J_j\in S_i}p_{jk}(k=1,2,\cdots,l)$$

对于两台平行机带约定时间的多维任务负载平衡问题，问题的目标是最大化 X。

$$X=\sum_{i=1}^{2}\min\{L_{i\max},d\}$$

下面给出问题 $P_2\,|\,d_j=d,VC\,|\max(X)$ 的一个实例：M_1、M_2 表示两台平行机，表 1 给出 3 项任务对应的向量。在这里，任务维数 $l=3$，约定时间 $d=5$。根据目标函数的定义，该问题的最优解是把任务 J_1 和 J_3 分配给 M_1，任务 J_2 分配给 M_2，最优值目标函数值为 8。

Table 1. Example of multidimensional task scheduling with a common due date
表 1. 带约定时间的多维负载平衡实例

	任务 J_1	任务 J_2	任务 J_3
p	(2, 1, 1)	(1, 5, 2)	(1, 0, 1)
d	5	5	5

3. $(\sqrt{5}-1)l$-在线算法

在这一部分，本文根据实例 $I=(M,J,p,d)$，设计了一个辅助实例 $\hat{I}=(M,J,\hat{p},d)$。对于实例 I，任务 J_j 用 l-维向量 $p_j=(p_{j1},p_{j2},\cdots,p_{jl})^{\mathrm{T}}$ 描述。对于实例 $\hat{I}$，任务 J_j 用 1 维向量 $\hat{p}_j=\sum_{k=1}^{l}p_{jk}$ 描述。根据实例 I 的目标函数定义，实例 $\hat{I}$ 目标函数是最大化 $\sum_{i=1}^{2}\min\left\{\sum_{j\in S_i}\hat{p}_j,d\right\}$。

在线算法的设计策略如下：我们用 EFF-LPT [1]算法分配实例 $\hat{I}$ 中的任务，可得实例 $\hat{I}$ 的一个可行解 (S_1,S_2)。根据实例 $\hat{I}$ 和 I 的定义可知，(S_1,S_2) 也是 I 的一个可行解。对于可行解 (S_1,S_2)，我们分别用 $OUT(\hat{I})$、$OUT(I)$ 表示实例 $\hat{I}$ 与 I 的输出值。$OPT(\hat{I})$ 与 $OPT(I)$ 则分别表示实例 $\hat{I}$ 与 I 的最优值。

MT-LPT **在线算法**

Step1：对于任意输入实例 I，设计辅助实例 $\hat{I}$；

Step2：运用 EFF-LPT [1]算法求解实例 I，可得实例 I 的一个可行解 (S_1,S_2)；

Step2.1：任务 $J_j(j=1,\cdots,n)$ 到达时，根据 p_j 构造 $\hat{p}_j$。

Step2.2：$i=1,i\le 2,i++$；

如果 $t(S_i^{j-1})+\hat{p}_j\le(\sqrt{5}-1)l$，分配 J_j 给机器 M_i。

Step2.3：否则，分配 J_j 给负载小的机器。

Step3：对于 (S_1,S_2)，分别输出实例 I 与 $\hat{I}$ 的目标值 $OUT(I)$、$OUT(\hat{I})$。

定理 1 对于问题 $P_2\,|\,d_j=d,VC\,|\max(X)$，在线算法 MT-LPT 的竞争比为 $(\sqrt{5}-1)l$。

证明：我们由文[1]可得，平行机的台数为 2 时，用 MT-LPT 求解实例 $\hat{I}$，则有

$$OPT\left(\hat{I}\right)\le\left(\sqrt{5}-1\right)OUT\left(\hat{I}\right) \tag{1}$$

假设 $OUT(I)=\min\{L_{1\max},d\}+\min\{L_{2\max},d\}$，由实例 I 目标值的定义，则

$$lOUT(I)\ge\min\left\{\sum_{J_j\in S_1}\sum_{k=1}^{l}p_{jk},d\right\}+\min\left\{\sum_{J_j\in S_2}\sum_{k=1}^{l}p_{jk},d\right\} \tag{2}$$

由实例 $\hat{I}$ 目标值的定义，则

$$OUT\left(\hat{I}\right)=\min\left\{\sum_{J_j\in S_1}\sum_{k=1}^{l}p_{jk},d\right\}+\min\left\{\sum_{J_j\in S_2}\sum_{k=1}^{l}p_{jk},d\right\} \tag{3}$$

联立(2)和(3)式，则有

$$OUT\left(\hat{I}\right)\le lOUT(I) \tag{4}$$

接下来，本文用反证法证明 $OPT(I)\le OPT\left(\hat{I}\right)$ 成立。

假设 $OPT(I)>OPT\left(\hat{I}\right)$。

令 $\left(S_1^*,S_2^*\right)$ 表示实例 I 的最优解。由实例 I 目标值的定义可知，

$$\begin{aligned}OPT(I)&=\min\{L_{1\max},d\}+\min\{L_{2\max},d\}\\&\le\min\left\{\sum_{J_j\in S_1^*}\sum_{k=1}^{l}p_{jk},d\right\}+\min\left\{\sum_{J_j\in S_2^*}\sum_{k=1}^{l}p_{jk},d\right\}\\&\le OPT\left(\hat{I}\right)\end{aligned}$$

这与假设矛盾。因此，则有

$$OPT(I)\le OPT\left(\hat{I}\right) \tag{5}$$

由(1)、(4)、(5)式，则有

$$OUT(I)\le\left(\sqrt{5}-1\right)lOPT(I) \tag{6}$$

由以上证明可知，对于带约定时间的多维任务负载平衡问题，MT-LPT 的竞争比为 $\left(\sqrt{5}-1\right)l$。

4. 问题 $P_2\mid d_j=d,VC\mid\max(X)$ 的动态规划算法

我们设计了一个动态规划算法 DY-AL 求解问题 $P_2\mid d_j=d,VC\mid\max(X)$。算法思想如下：把 n 项任务所有的可行解尝试一遍，从而求出 $P_2\mid d_j=d,VC\mid\max(X)$ 的最优解。下面给出具体的算法步骤和一些符号说明。

$\varepsilon_i(i=1,2)$ 是第 i 列元素为 1，其它的元素是 0 的 $l\times 2$ 维矩阵。我们定义一个新的运算，令 $p_j\otimes\varepsilon_i$ 为列向量 p_j 每个分量和 ε_i 的第 i 列元素对应相乘。

举个例子 $p_1\otimes\varepsilon_1=\begin{pmatrix}p_{11}\\p_{12}\\\vdots\\p_{1l}\end{pmatrix}\otimes\begin{pmatrix}1&0\\1&0\\\vdots&\vdots\\1&0\end{pmatrix}=\begin{pmatrix}p_{11}&0\\p_{12}&0\\\vdots&\vdots\\p_{1l}&0\end{pmatrix}$。

令 $\Psi_j(j=0,1,\cdots,n)$ 表示加工完前 j 项任务后所有可能的负载矩阵构成的集合。

DY-AL **动态规划算法**

Step1：置 $\Psi_0=\{(0)_{l\times 2}\}$；

Step2：当任务 $J_j\left(j=1,\cdots,n\right)$ 到达时，集合 Ψ_j 可由 Ψ_{j-1} 递推得到，计算方法如下：

$$\Psi_j=\Psi_{j-1}+\left\{p_j\otimes\varepsilon_i\,|\,i=1,2\right\};$$

Step3：对集合 Ψ_n 所有的负载矩阵，提取第一列的最大值得到集合 $\left\{c_1,c_2,\cdots,c_{|\Psi_n|}\right\}$，提取第二列的最大值得到集合 $\left\{\tilde{c}_1,\tilde{c}_2,\cdots,\tilde{c}_{|\Psi_n|}\right\}$，$|\Psi_n|$ 表示空间 Ψ_n 中负载矩阵的个数。令 $\alpha=\max\left\{c_1,c_2,\cdots,c_{|\Psi_n|}\right\}$，$\beta=\max\left\{\tilde{c}_1,\tilde{c}_2,\cdots,\tilde{c}_{|\Psi_n|}\right\}$。最后输出 $\min\left\{\alpha,d\right\}+\min\left\{\beta,d\right\}$。

由动态规划算法的步骤可知，问题 $P_2\,|\,d_j=d,VC\,|\max\left(X\right)$ 的最优解可由 DY-AL 得到。

5. 问题 $P_2\,|\,d_j=d,VC\,|\max\left(X\right)$ 的完全多项式时间近似方案

当任务维数为给定常数时，在动态规划的基础上采用舍入取整技术，设计了一个全多项式时间近似方案(FPTAS)。

定理 2 当 l 给定值时，问题 $P_2\,|\,d_j=d,VC\,|\max\left(X\right)$ 在 $O\left(n^{2l+1}\left(\frac{l^2}{\varepsilon}\right)^{2l}\right)$ 时间内存在一个 FPTAS。

证明：在这里，令 L 是 MT-LPT 处理实例 I 得到的输出值。由定理 1 可知，

$$L\le OPT\left(I\right)\le\left(\sqrt{5}-1\right)lL \tag{7}$$

根据实例 $I=\left(M,J,p,d\right)$ 设计辅助实例 $I'=\left(M,J,p',d\right)$。对于实例 I 来说，任务 J_j 与向量 $p_j=\left(p_{j1},p_{j2},\cdots,p_{jl}\right)^{\mathrm{T}}$ 对应。对于辅助实例 I' 来说，任务 J_j 与向量 $p'_j=\left(p'_{j1},p'_{j2},\cdots,p'_{jl}\right)^{\mathrm{T}}$ 对应。我们令 $\alpha=\sqrt{5}-1$，而且令 $p'_{jk}=\left\lfloor\frac{p_{jk}}{\varepsilon L/\alpha nl}\right\rfloor\cdot\frac{\varepsilon L}{\alpha nl}\left(k=1,2,\cdots,l\right)$。由 $p'_{jk}=\left\lfloor\frac{p_{jk}}{\varepsilon L/\alpha nl}\right\rfloor\cdot\frac{\varepsilon L}{\alpha nl}$ 的定义，我们可以得到 $p'_{jk}\le p_{jk}\le p'_{jk}+\frac{\varepsilon L}{\alpha nl}$。

假设 $\left(S'_1,S'_2\right)$ 是算法 DY-AL 求解实例 $I'=\left(M,J,p',d\right)$ 的最优解，则最优值

$$OPT\left(I'\right)=\min\left\{\max_k\sum_{j\in S'_1}p'_{jk},d\right\}+\min\left\{\max_k\sum_{j\in S'_2}p'_{jk},d\right\}。$$

假设实例 I 的最优解是 $\left(O_1,O_2\right)$，则最优值

$$OPT\left(I\right)=\min\left\{\max_k\sum_{j\in O_1}p'_{jk},d\right\}+\min\left\{\max_k\sum_{j\in O_2}p'_{jk},d\right\}。$$

依据实例 I' 的最优解 $\left(S'_1,S'_2\right)$ 调度实例 I，则有

$$\begin{aligned}
&\min\left\{\max_k\sum_{j\in S'_1}p'_{jk},d\right\}+\min\left\{\max_k\sum_{j\in S'_2}p'_{jk},d\right\}\\
&\le\min\left\{\max_k\sum_{j\in S'_1}\left(p'_{jk}+\frac{\varepsilon L}{\alpha nl}\right),d\right\}+\min\left\{\max_k\sum_{j\in S'_2}\left(p'_{jk}+\frac{\varepsilon L}{\alpha nl}\right),d\right\}\\
&\le\min\left\{\max_k\sum_{j\in S'_1}p'_{jk},d\right\}+\min\left\{\max_k\sum_{j\in S'_2}p'_{jk},d\right\}+n\frac{\varepsilon L}{\alpha nl}\\
&=\min\left\{\max_k\sum_{j\in S'_1}p'_{jk},d\right\}+\min\left\{\max_k\sum_{j\in S'_2}p'_{jk},d\right\}+\frac{\varepsilon L}{\alpha l}
\end{aligned}$$

$$\le \min\left\{\max_k \sum_{j\in O_1} p'_{jk}, d\right\} + \min\left\{\max_k \sum_{j\in O_2} p'_{jk}, d\right\} + \frac{\varepsilon L}{\alpha l}$$
$$\le \min\left\{\max_k \sum_{j\in O_1} p_{jk}, d\right\} + \min\left\{\max_k \sum_{j\in O_2} p_{jk}, d\right\} + \frac{\varepsilon L}{\alpha l}$$
$$\le OPT(I) + \varepsilon OPT(I)$$
$$= (1+\varepsilon) OPT(I)$$

下面分析算法的运行时间，由定理 2 中的(7)式可知，$OPT(I') \le OPT(I) \le \alpha l L$，所以算法 DY-AL 求解实例 I' 时，只计算负载分量不超过 $\alpha l L$ 的可行解，而且机器的各负载分量又是 $\frac{\varepsilon L}{\alpha n l}$ 的整数倍，则机器负载最多有 $\alpha l L \Big/ \frac{\varepsilon L}{\alpha n l} + 1 = \frac{\alpha^2 n l^2}{\varepsilon} + 1$ 种可能。又因为每个负载矩阵的元素个数为 $2l$，而且任务总个数为 n，所以实例 I' 能够在 $O\left(n\left(\frac{\alpha^2 n l^2}{\varepsilon}+1\right)^{2l}\right) = O\left(n^{2l+1}\left(\frac{l^2}{\varepsilon}\right)^{2l}\right)$ 时间找到最优解。

6. 结论

本文研究了两台平行机上带约定时间的多维任务负载平衡问题，设计了一个在线算法，算法的竞争比为 $\left(\sqrt{5}-1\right)l$，然后给出该问题的一个 FPTAS。对于带约定时间的多维任务负载平衡问题，以后我们希望设计一个近似算法。

基金项目

楚雄师范学院研究项目(XJYB2004)资助。

参考文献

[1] Chen, X., Sterna, M., Han, X., *et al*. (2016) Scheduling on Parallel Identical Machines with Late Work Criterion: Offline and Online Cases. *Journal of Scheduling*, **19**, 729-736. https://doi.org/10.1007/s10951-015-0464-7

[2] Chen, X., Kovalev, S., Liu, Y., *et al*. (2021) Semi-Online Scheduling on Two Identical Machines with a Common Due Date to Maximize Total Early Work. *Discrete Applied Mathematics*, **290**, 71-78. https://doi.org/10.1016/j.dam.2020.05.023

[3] Xiao, M., Liu, X. and Li, W. (2021) Semi-Online Early Work Maximization Problem on Two Hierarchical Machines with Partial Information of Processing Time. In: Wu, W. and Du, H., eds., *Algorithmic Aspects in Information and Management*, Springer, Cham, 146-156. https://doi.org/10.1007/978-3-030-93176-6_13

[4] Graham, R.L., Lawler, E.L., Lenstra, J.K., *et al*. (1979) Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey. *Annals of Discrete Mathematics*, **5**, 287-326. https://doi.org/10.1016/S0167-5060(08)70356-X

[5] Chekuri, C. and Khanna, S. (2004) On Multidimensional Packing Problems. *SIAM journal on Computing*, **33**, 837-851. https://doi.org/10.1137/S0097539799356265

[6] Meyerson, A., Roytman, A. and Tagiku, B. (2013) Online Multidimensional Load Balancing. In: Raghavendra, P., Raskhodnikova, S., Jansen, K. and Rolim, J.D.P., eds., *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, Springer, Berlin, Heidelberg, 287-302. https://doi.org/10.1007/978-3-642-40328-6_21

[7] Im, S., Kell, N., Kulkarni, J., *et al*. (2015) Tight Bounds for Online Vector Scheduling. 2015 *IEEE 56th Annual Symposium on Foundations of Computer Science*, Berkeley, CA, USA, 17-20 October 2015, 525-544. https://doi.org/10.1109/FOCS.2015.39

[8] Bansal, N., Oosterwijk, T., Vredeveld, T., *et al*. (2016) Approximating Vector Scheduling: Almost Matching Upper and Lower Bounds. *Algorithmica*, **76**, 1077-1096. https://doi.org/10.1007/s00453-016-0116-0

[9] Li, W. and Cui, Q. (2018) Vector Scheduling with Rejection on a Single Machine. *4OR*, **16**, 95-104. https://doi.org/10.1007/s10288-017-0356-0

[10] Dai, B. and Li, W. (2020) Vector Scheduling with Rejection on Two Machines. *International Journal of Computer Mathematics*, **97**, 2507-2515. https://doi.org/10.1080/00207160.2019.1711373