在线带测试多处理器调度问题竞争比下界研究

张志心 3210106357
浙江大学

指导教师：谈之弈

2025 年 2 月 19 日

多处理器调度问题/平行机调度问题[1]是最早被广泛研究的 NP-Hard 组合优化问题之一。它研究 n 个作业在 m 台并行相同的机器集上非抢占执行的最小完工时间。带测试调度问题[2][3][4]是该问题的变体，具有广泛的研究意义。

问题 (带测试多处理器调度问题)

一个实例 I 包含一组 n 个作业 $J = \{J_1, J_2, \cdots, J_n\}$，每个作业都要在 m 台并行相同的机器集 $M = \{M_1, M_2, \cdots, M_m\}$ 上非抢占执行。每个作业带有一个处理时间上界 u_j 和一个测试操作的时长 t_j，如果进行测试，作业将得到一个实际运行时长 p_j。每一个作业 J_j 都可以选择在某台机器上花 u_j 时间运行，或者先花 t_j 时间进行测试，之后在同一台机器上运行 p_j 时间。目标是最小化完工时间 C_{max}。

[1] Gary M R, Johnson D S. Computers and Intractability: A Guide to the Theory of NP-completeness[J]. 1979.
[2] Dürr C, Erlebach T, Megow N, et al. Scheduling with explorable uncertainty[C] (ITCS 2018). Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2018.
[3] Dürr C, Erlebach T, Megow N, et al. An adversarial model for scheduling with testing[J]. Algorithmica, 2020, 82(12): 3630-3675.
[4] Albers S, Eckl A. Scheduling with testing on multiple identical parallel machines[C] WADS 2021, Virtual Event, August 9–11, 2021, Proceedings 17. Springer International Publishing, 2021: 29-42.

- **半在线问题**：所有作业在时间零时刻到达，记作 $P \mid t_j, 0 \le p_j \le u_j \mid C_{\mathsf{max}}$。
- **在线问题**：调度器需要在作业到达时做出测试决策，并指定机器进行测试或直接执行，每个作业在**上一个作业的调度决策完成后**到达，记作 $P \mid \mathsf{online}, t_j, 0 \le p_j \le u_j \mid C_{\mathsf{max}}$。

调度器应当利用作业到达时已知的信息，决定是否对作业进行测试，从而尽可能平衡由于处理时间未知所带来的总时间开销。

算法的性能可以通过竞争比来衡量。

定义 (竞争比)

假设存在一个确定性算法，令 $C(I)$ 表示该算法在实例 I 上产生的完工时间，$C^*(I)$ 表示最优离线调度的完工时间。定义该算法的竞争比为

$$\alpha = \sup_I \frac{C(I)}{C^*(I)}, \quad \text{其中}I\text{遍历所有问题实例},$$

该算法被称为 α-竞争算法。

注： 对于带测试多处理器调度的离线最优算法，可以视为每个工作的处理时间为 $u'_j = \min\{u_j, t_j + p_j\}$ 的普通多处理器调度问题。

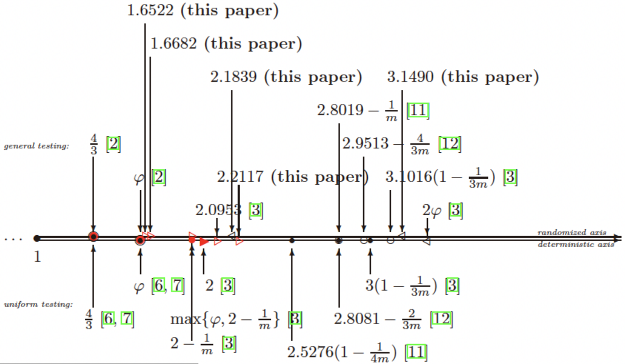
通用测试实例：对于所有测试操作时间都为单位时间的问题 $P \mid \text{online}, t_j = 1, 0 \le p_j \le u_j \mid C_{\max}$。

- 当只有一台机器时，Dühr 等[1] [2] 提出，当 $u_j \le \varphi := \frac{\sqrt{5}+1}{2}$ 时对作业 j 进行测试，得到 φ-竞争算法；或者以概率 $f(u_j) = \max\left(0, \frac{u_j(u_j-1)}{u_j(u_j-1)+1}\right)$ 概率对作业进行测试，得到 $\frac{4}{3}$-竞争的随机化算法。可以证明[3]，这两种算法是最优的，即 φ 是确定性算法的竞争比下界，$\frac{4}{3}$ 是任何随机化算法的期望竞争比下界。
- 当有 $m \ge 2$ 台机器时，存在 $\varphi(2 - \frac{1}{m})$-竞争算法 [4]。
- 对于任何确定性算法，**竞争比下界为** 2。（构造实例用了两台机器、三个工件）[5]

[1] Dürr C, Erlebach T, Megow N, et al. Scheduling with explorable uncertainty[C] (ITCS 2018). Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2018.
[2] Dürr C, Erlebach T, Megow N, et al. An adversarial model for scheduling with testing[J]. Algorithmica, 2020, 82(12): 3630-3675.
[3] Albers S, Eckl A. Scheduling with testing on multiple identical parallel machines[C] WADS 2021, Virtual Event, August 9–11, 2021, Proceedings 17. Springer International Publishing, 2021: 29-42.
[4] Graham R L. Bounds for certain multiprocessing anomalies[J]. Bell system technical journal, 1966, 45(9): 1563-1581.
[5] Albers S, Eckl A. Scheduling with testing on multiple identical parallel machines[C] WADS 2021, Virtual Event, August 9–11, 2021, Proceedings 17. Springer International Publishing, 2021: 29-42.

在 Gong 的论文[1] 中，证明了任何**确定性**算法的竞争比下界为 2.2117。

同时，论文还提出了一种确定性算法集成的随机化算法，其期望竞争比为 $\frac{3\varphi+3\sqrt{13-7\varphi}}{4} \approx 2.1839$。



[1] Gong M, Chen Z Z, Lin G, et al. Randomized algorithms for fully online multiprocessor scheduling with testing[J]. arXiv preprint arXiv:2305.01605, 2023.

主要研究内容

探索并证明在线带测试的多处理器调度问题的**确定性算法**竞争比下界。即：证明不存在竞争比总是低于某个下界的确定性算法。

$$P \mid \text{online}, t_j, 0 \leq p_j \leq u_j \mid C_{\max}$$

对该问题进行程序建模，通过分析特定情况下的竞争比表现，识别出影响竞争比的关键因素。在此基础上，我还将对现有算例进行分析，特别关注那些导致竞争比达到最坏的特殊情况。运用计算机辅助的方法在**有限**的时间内搜索出更广泛的实例，以验证该问题的竞争比下界，并通过严谨的分析来确认最终结果。

研究意义和目的

1. 多处理器调度问题作为一个重要的组合优化问题，其**在实际生产环境中具有广泛的应用场景**，如工业生产和资源分配等领域。
2. **带测试的多处理器调度问题**，引入了相同的工作的不同处理方式（一种未知效果的新处理手段），并以提高整体系统的效率为目标，同样有着重要的研究意义。
3. 目前在线带测试多处理器调度问题竞争比下界还有较大的提升空间。现有的研究主要集中在**机器数量、工作数量较少的情况下，使用手动构造最坏情况**来分析竞争比下界，但对于更一般的情形，特别是当 $m \geq 3, n \geq 4$ 时，现有的竞争比下界仍有改进余地。
4. 通过引入计算机辅助方法，利用程序设计方法**更高效地构建和分析决策树**，可以扩大搜索范围，有希望进一步提高竞争比下界。

同时我还将参照朴素平行机调度问题[2] 数学模型，设计**带测试的多处理器调度问题的竞争比下界求解模型**。

竞争比下界求解可以看成是一个 max-min 的优化问题。

1. 设决策序列集合为 A，实例集合为 $\mathcal{I}$，则最坏情况下的竞争比可表示为 $\alpha^* = \sup_{I\in\mathcal{I}} \alpha(I) = \sup_{I\in\mathcal{I}} \min_{a\in A} \frac{C(I,a)}{C^*(I)}$。
2. 将最优化问题转为判定性问题：给定 α_0，判断是否存在实例 I^* 使得 $\alpha(I) \geq \alpha_0$，即 $\min_{a\in A} \frac{C(I,a)}{C^*(I)} \geq \alpha_0$，即 $\forall a \in A, C(I,a) \geq \alpha_0 C^*(I)$。
3. 不等号左右两端都是 I 的分量的线性组合及其 min 和 max。问题转化为一个或多个线性规划问题，可以通过加以剪枝的搜索方法来构造每个分量的值。

[2] Gormley T, Reingold N, Torng E, et al. Generating adversaries for request-answer games[C]//Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms. 2000: 564-565.

研究计划进度安排及预期目标

进度安排

1. 2025.2.28 以前，完成问题建模和文献调研，归纳现有方法。完成三合一等准备工作。
2. 2025.3.15 以前，设计计算机搜索的代码并初步验证现有研究成果。
3. 2025.3.31 以前，通过进一步的理论分析，确定搜索剪枝策略。
4. 2025.4.15 以前对剪枝策略和现有运行结果，进行深入理论分析和证明，进一步改进竞争比下界。
5. 2025.5 以前，撰写毕业论文，持续尝试改进搜索策略，以得到更好的结果和获得更快的程序验证效率。

预期目标

1. 通过广泛实例模拟，验证现有研究成果中的竞争比下界。
2. 通过计算机进行高效剪枝搜索，寻找更大的竞争比下界。