arXiv:2611.32688v1 [cs.DS] 24 Nov 2026

# 连续单调可塑作业调度

Bennet Edler* Klaus Jansen† Felix Ohnesorge‡

**摘要**

本文研究连续单调可塑作业调度（CMMJS）问题：一组作业需要在并行机器上进行调度，以最小化整体最大完工时间。每个作业由一个处理时间函数刻画，该函数将分配的机器数量映射为处理时间，且相应的工作区域函数单调不减。基于 Jansen 和 Ohnesorge（2026）提出的框架及识别出的关键情形，我们给出一种改进的分析方法，将近似比收紧至 $\frac{2\sqrt{e}}{2\sqrt{e}-1} + \varepsilon \approx 1.4353 + \varepsilon$，运行时间为 $O(nm \log \frac{1}{\varepsilon})$。我们通过给出一个匹配的下界实例，证明这种最坏情况分析基本上是紧的。该近似比不仅适用于连续调度，也适用于该问题的非连续变体。最后，我们证明改进后的界限可以自然地推广至异构环境；将我们的结果无缝整合到 Germann 等人（2024）提出的 CPU/GPU 联合调度现有双重近似框架中，可将联合问题的整体近似比改进至 $\frac{2\sqrt{e}}{2\sqrt{e}-1} + \varepsilon$。

**关键词**。 调度、近线性时间算法、近似。

**资助**。 德国研究基金会（DFG）– 项目编号453769249。

## 1. 引言

在连续单调可塑作业调度（CMMJS）中，给定 $n \in \mathbb{Z}_{>0}$ 个作业和$m \in \mathbb{Z}_{>0}$ 台机器。每个作业 $j \in [n]$ 都与一个函数 $t_j$ 相关联，该函数 $[m] \to \mathbb{Q}_{>0}$ 将机器数量映射为处理时间。此外，工作区域函数$w_j : [m] \to \mathbb{Q}_{>0}$，其中$w_j(k) := t_j(k) \cdot k$，定义了作业 $j$在分配到 $k$ 台机器时的工作区域。我们假设$w_j$ 是非递减的，而 $t_j$ 是非递增的。调度通过为每个作业$j$ 分配一定数量的机器 $\alpha_j \in [m]$ 和一个开始时间$s_j$来确定。在任何可行调度中，任意时刻分配给作业的机器总数不得超过 $m$。如果对于每个作业 $j$ ，分配给它的机器都是连续的，则称该调度为连续调度，即如果 $j$ 被分配到机器 $m_j, \ldots, m_j + \alpha_j - 1$，其中 $m_j \in [m - \alpha_j + 1]$。我们的目标是最小化调度的最大完工时间，即 $\max_{j\in[n]} s_j + t_j(\alpha_j)$。我们将最小最大完工时间记为OPT。对于任意机器$i \in [m]$，如果某个作业曾在该机器上执行，则称该机器的负载为该作业执行的最晚时间

机器，即负载$(i) := \max_{j\in[n],\ i\in[m_j, m_j+\alpha_j)} s_j + t_j(\alpha_j)$。

*基尔大学，stu235896@mail.uni-kiel.de，https://orcid.org/0009-0002-5067-6781
†基尔大学，kj@informatik.uni-kiel.de， https://orcid.org/0000-0001-8358-6796
‡基尔大学，foh@informatik.uni-kiel.de， https://orcid.org/0009-0003-8023-3380

˜表1：单调可塑作业调度的实用高效算法。该 $O$ 记号省略了 $n$、 $m$和 $1/\varepsilon$中的对数因子。带 **(*)** 标记的算法最初是为非连续情形设计的，但可以调整为适用于连续情形 [JO26]。

| 比例 | 运行时间 | 连续? | 作者 |
|---|---|---|---|
| $\frac{3}{2}+\varepsilon$ | $O(nm$ 对数$(1/\varepsilon))$ | Yes* | Mounié、Rapine、Trystram（2007年） [MRT07] |
| $\frac{3}{2}+\varepsilon$ | $\tilde{O}(n/\varepsilon^3)$ | Yes* | Jansen, Land（2018年） [JL18] |
| $\frac{3}{2}+\varepsilon$ | $O(n/\varepsilon)$ | Yes* | Grage, Jansen, Ohnesorge (2023) [GJO23] |
| $\frac{3}{2}$ | $O(nm)$ | No | Wu、Zhang、Chen (2023) [WZC23] |
| $1.4593+\varepsilon$ | $O(nm$ 对数$(1/\varepsilon))$ | Yes | Jansen, Ohnesorge（2026） [JO26] |
| $1.4353+\varepsilon$ | $O(nm$ 对数$(1/\varepsilon))$ | Yes | **本文工作** |

众所周知，一般的可塑作业调度问题是 NP 困难的 [DL89]，并且若不假设单调性，则不存在多项式时间（$\frac{3}{2}-\varepsilon$）近似算法，除非 P $=$NP [Dro95; Joh06]。在特殊假设下，存在更好的多项式时间近似算法 [DLM06; JP99]，以及一种伪多项式（$\frac{5}{4}+\varepsilon$）近似算法，其运行时间为$O(n\log\ n)\cdot m^{f(\frac{1}{\varepsilon})}$ [JR19],，其中 $f(\cdot)$ 是可计算函数。当机器数量 $m$ 受限于关于 $n$ 的多项式时，Jansen 和 Thöle 针对非连续变体提出了多项式时间近似方案（PTAS） [JT08]，并针对连续变体提出了一个（$\frac{3}{2}+\varepsilon$）近似算法。对于这一问题，在没有特殊限制的情况下，存在2-近似算法 [TWY92; LT94]。

当假设处理时间函数 $t_j$ 是单调的时，该问题仍然是 NP-困难的 [JLL16],，但严格的 3 2 下界不再适用。Jansen 和 Land 针对连续变体提出了全多项式时间近似方案（FPTAS），条件为 ($m>8^n_\varepsilon$) [JL18]，并据此将非连续变体的 PTAS [JT08]推广到该情形；然而，即使对于 $\varepsilon=\frac{1}{2}$，该 PTAS 的复杂度也大致为 $nm^{2\cdot10^{2.4\cdot10^{17}}}$。对于伪多项式时间 ($\frac{5}{4}+\varepsilon$)-近似算法 [JR19],，也可以得出类似结论：当 $\varepsilon=\frac{1}{4}$ 时，其时间复杂度为 $\Omega((mn)^{16^{4^{+3}}})$。这些较大的常数促使近期研究转向关注实际高效的算法，即运行时间为 $O(nm\log(\frac{1}{\varepsilon}))$ 或更优的算法。

Mounié、Rapine 和 Trystram 提出了一种 ($\frac{3}{2}+\varepsilon$)-近似算法 [MRT07]。Jansen 和 Land [JL18] 以及 Grage 等人 [GJO23]随后改进了该算法的复杂度。尽管原始工作中未明确说明，这些算法也可以调整以适用于连续情形 [JO26]。Wu、Zhang 和 Chen [WZC23] 消除了近似比中的 $\varepsilon$因子，实现了一个 3_2-近似算法。就在最近，Jansen 和 Ohnesorge [JO26] 提出了一种适用于连续变体、最坏情况比率约为 $1.4593+\varepsilon$ 的算法。请注意，运行时间 $O(nm)$严格来说仅为伪多项式时间；但由于 Jansen 和 Land [JL18]在机器数量较大时（$m>8^n_\varepsilon$）开发了一种 FPTAS，我们可以假设 $m$ 在 $n$ 方面是多项式的，从而使运行时间达到多项式时间。

设计这一单调变体的算法时，也许最有用的观察是：减少分配给某个作业的机器数量并不会增加工作区域。这一点已被 Jansen 和 Land [JL18] 用于对分配到许多机器的作业进行舍入。

**定义 1**。对于任意 $j\in[n]$ 和 $\alpha'<\alpha\le m$，单调性意味着：

$$t_j(\alpha')\le\frac{w_j(\alpha)}{\alpha'}=t_j(\alpha)\frac{\alpha}{\alpha'}. \tag{1}$$

例如，wh 我们将一个作业分配给一台而不是两台机器时，新的处理时间最多会 l

增加一倍。与先前的工作类似 [GJO23； JO26； MRT07], 我们将高度依赖这一性质来证明我们的主要结果：

**定理 2**。 令 $\varepsilon > 0$。对于连续单调可塑作业调度，存在一种近似比为 $\frac{2\sqrt{e}}{2\sqrt{e}-1} + \varepsilon$、运行时间为 $O(nm \log \frac{1}{\varepsilon})$的算法。

值得注意的是，尽管我们的算法是针对连续情形设计的，但对于非连续变体，它仍能达到完全相同的近似比。这一性质并非显然成立，因为非连续调度的最优最大完工时间可能严格小于连续调度的最优最大完工时间。然而，我们的分析——类似于其他近期连续算法所采用的方法 [GJO23； JO26]——使用最大总工作面积来界定所得的最大完工时间。因此，我们的近似比 $\frac{2\sqrt{e}}{2\sqrt{e}-1} + \varepsilon$ 同样适用于非连续情形。

为补充这一上界，我们还在附录 B中提供了一个紧下界示例，证明该算法的最坏情况分析是紧的：

**引理 3**。 对于任意 $\varepsilon > 0$，存在一个实例 $I$，使得算法 *1* 返回的调度的最大完工时间为 $\mathrm{ALG}(I)$，且 $\frac{\mathrm{ALG(I)}}{\mathrm{OPT}(I)} \geq \frac{2\sqrt{e}}{2\sqrt{e}-1} - \varepsilon$。

此外，我们证明，针对连续单调可塑作业调度所得到的改进界可以自然地扩展到异构环境。具体而言，我们的结果可直接应用于现有的 CPU/GPU 调度问题算法 [GJOT24]。在该问题中，我们同样给定一个 $n$ 作业集合，但每个作业将要么在 GPU 上执行，要么在 CPU 上执行。在 GPU 上，一个作业可以在单台机器上执行，并具有给定的处理时间（这对应于$P||C_{max}$）；而在 CPU 上，一个作业具有如 CMMJS 中所述的处理时间函数。现在的任务是将作业划分为两个集合，并分别为这两个集合制定调度方案，使整体最大完工时间最小。

尽管 GPU 调度问题（$P\|C_{\max}$）自 20 世纪 60 年代以来一直受到持续研究 [Gra69；HS87；Jan10；JGJ78], ，但该问题与更复杂的CMMJS问题的结合，最早由 Bleuse 等人研究。 [BKM+15]。他们提出了一种（$\frac{3}{2} + \varepsilon$）-近似算法，其运行时间为 $O(n^2 m_c^4 m_g^2)$。之后，Germann 等人 [GJOT24] 将运行时间改进为 $O(\log(nm_c m_g/\varepsilon) \cdot n^2 m_c m_g)$，同时保持了相同的近似比。我们证明，针对CMMJS 所得的结果可以无缝整合到他们的算法中，从而改进 CPU/GPU 联合调度问题的近似比。

通过调整 [GJOT24] 中的松弛参数，使其与我们更紧的 CPU 界精确匹配，我们的分析可以无缝接入其多维多重选择背包问题（MMCKP）表述。这一直接的代换改进了他们的双重近似框架，使 CPU/GPU 联合调度问题的整体近似比提升至 $\frac{2\sqrt{e}}{2\sqrt{e}-1} + \varepsilon$。

**定理 4**。 设 $\varepsilon > 0$。存在一种用于 *CPU/GPU* 调度的（$\frac{2\sqrt{e}}{2\sqrt{e}-1} + \varepsilon$）-近似算法，其运行时间为 $O$（log（$nm_c m_g/\varepsilon$）· $n^2 m_c m_g$）。

**论文结构** 在第 2节中，我们证明主要结果，即对 [JO26] 中所述算法进行改进分析。作为分析的补充，我们在附录 B 中给出一个下界示例，证明该算法无法达到优于 $\frac{2\sqrt{e}}{2\sqrt{e}-1}$ 的近似比。最后，在第 3节中，我们展示这一结果如何直接导出适用于 CPU/GPU 调度扩展的改进算法。

### 1.1. 预备知识

由于我们改进分析了 [JO26]，中的算法，因此简要概述其方法并建立所需的符号。利用 Hochbaum 和 Shmoys 开创的双重近似框架 [HS87]，，我们可以假设给定一个最大完工时间猜测 $d$，使得$\mathrm{OPT} \le d < (1+\varepsilon)\mathrm{OPT}$。为简单起见，我们还假设 $d = 1$（通过缩放）。

鉴于此， [JO26] 中的算法将作业划分为一组大作业 $J_B := \{j \in [n] \mid t_j(1) > \frac{3}{7}\}$ 和一组小作业 $J_S = [n] \setminus J_B$，并首先对大作业进行调度。该算法的关键组成部分是采用多重选择背包问题（MCKP）建模，将大作业划分为三个集合。我们在引理5中重新给出该划分的定义以及改进分析的结果，其中$\gamma_j(x)$表示满足其处理时间至多为 $x$的作业 $j$ 的最小机器分配，即$\gamma_j(x) = \min\{\alpha \in [m] \mid t_j(\alpha) \le x\}$。

**Lem理5** （改进 [JO26]）。 给定作业的一个划分 $C_1 \sqcup C_2 \sqcup C_3 = J_B$ $s.t.$

1. $\sum_{j\in C_1} w_j(\gamma_j(1)) + \sum_{j\in C_2} w_j(\gamma_j(\frac{4}{7})) + \sum_{j\in C_3} w_j(\gamma_j(\frac{3}{7})) \le m - \sum_{j\in J_S} t_j(1)$
2. $\sum_{j\in C_1} \gamma_j(1) + \frac{1}{2}\sum_{j\in C_2} \gamma_j(\frac{4}{7}) \le m,$

可以在时间 $O(nm)$ 内计算出最大完工时间至多为 $\frac{2\sqrt{e}}{2\sqrt{e}-1}$ 的连续调度。

注意，引理 5 所需的划分可以通过多重选择背包问题找到。算法 1 概述了如何在 [JO26]中调度作业。为完整起见，我们给出

**算法 1** CMMJS近似算法 [JO26]

**要求：** 划分 $C_1 \sqcup C_2 \sqcup C_3 = J_B$ 和小作业 $J_S$

1: 按照 $S_0$、 $S_1$、 $S_2$ 从 $C_1$、 $C_2$、 $C_3$ 构造，如 [JO26]所示。▷算法 m 4
2: 在时间 0 将作业调度到 $S_0 \sqcup S_1$ 中，使机器的负载非递减。3: 使用算法 3或算法 2 将作业调度到 $S_2$ 中。4: 贪心地调度小作业 $J_S$。

构造 $S_0$、 $S_1$和 $S_2$ 的过程见附录 E （算法 4）。此处，我们仅给出总体思路，并在引理 7中总结这些集合的重要性质。

备注 6（调度）。对于给定的目标最大完工时间 $\lambda \in (1, 3\,2)$，原算法将每个作业 $j \in S_0$ 分配到 $\alpha_j := \gamma_j(\lambda)$ 台机器、每个作业 $j \in S_1$ 分配到 $\alpha_j := \gamma_j(1)$ 台机器，并将每个作业 $j \in S_2$ 分配到$\alpha_j := \gamma_j((\lambda - 1))$ 台机器，如图 1b 所示。下文中，我们将$S_0$ 和 $S_1$ 中的作业视为已分配到机器，并使其处理时间呈非递减顺序。注意，这样一来，某些机器可能没有安排任何单个作业。我们用 $q \ge 0$ 表示没有安排作业的机器数量。这如图 1a 所示。目标是找到最小的 $\lambda$，使得我们始终可以压缩 $S_2$ 中的作业，从而使调度的最大完工时间至多为 $\lambda$。

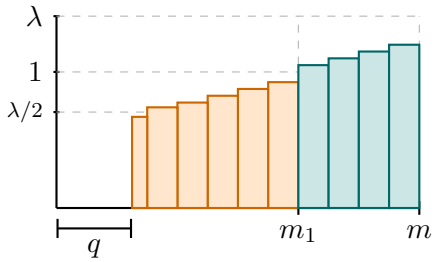
令 $m_1 := m - \sum_{j\in S_0} \gamma_j(\lambda)$ 表示放置 $S_0$后剩余的机器数量。根据构造， $S_0$ 中的每个作业的处理时间都严格大于 1。因此， $S_1$ 和 $S_2$ 中作业的总工作面积上界为 $m_1$。这些变换的结果是所谓的三层货架调度，其具有以下性质：

**引理 7** （[JO26],定义 9 和引理 10）。 给定 $\lambda \in [\frac{10}{7}, \frac{3}{2})$ 以及 $C_1 \sqcup C_2 \sqcup C_3 = J_B$ 的一个划分，如引理 5中所述，我们可以返回一个最大完工时间至多为 $\lambda$ 的连续调度，所需时间为$O(mn)$，或者构造一个划分 $S_0 \sqcup S_1 \sqcup S_2 = J_B$ 使得：

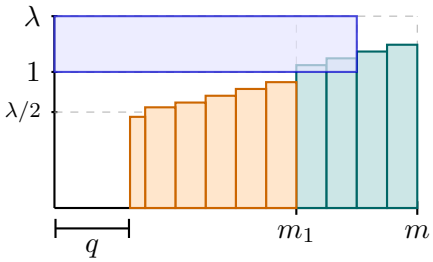
*1. a)* 总工作面积至多为 $\sum_{j\in S_1} w_j\,(\gamma_j\,(1)\,+\sum_{j\in S_2} w_j\,(\gamma_j\,(\lambda-1)\,\leq m_1$

*2. a)* $S_1$ 中的所有作业都可以调度到 $m_1-q$ 台机器上，最大完工时间至多为1。*b)* 除了分配到单台机器上的一个作业外， $S_1$ 中的每个作业的处理时间都大于$\frac{\lambda}{2}$ 。*c)* $S_1$ 的总工作面积至少为 $W_1:=\sum_{j\in S_1} w_j(\gamma_j(1))\geq \frac{\lambda}{2}(m_1-q)$。

*3. a)* $j\in S_2$ 中的每个作业的工作面积至少为$w_j(\gamma_j((\lambda-1)))>\lambda q\geq\lambda$ （因为 $q\geq 1$）。




(a) 绿色作业位于 $S_0$中，橙色作业位于 $S_1$中。




(b) 蓝色作业位于 $S_2$中

图 1：（a）如注释 6 所示的调度。（b）蓝色作业无法放置在橙色作业上方，因为其机器需求大于 $m_1$，因此与 $S_0$相交。

## 2. 连续单调可塑作业调度的更佳分析

在本节中，我们将假设引理 7 给出的性质成立，因为否则我们已经可以找到一个最大完工时间至多为 $\lambda$ 的调度。现在，我们将按照算法 $S_0\sqcup S_1$ 1 将作业放置在 $S_2$ 中，并说明如何处理 $S_2$。

**引理 8。** 设 $\lambda\geq\frac{2\sqrt{e}}{2\sqrt{e}-1}$。如果 $|S_2|>1$，那么我们可以创建一个最大完工时间至多为 $\lambda$ 的连续调度。

证明。我们将该引理的证明推迟到附录A，因为它类似于 [JO2 6]. □

利用上述引理，在剩余的分析中我们可以假设 $|S_2|=1$。我们关注 $S_2$ 中的作业，并将其称为 $\tilde{\jmath}$，将其工作区域记为 $W_2$。与 [JO26], 一样，我们使用算法 $\tilde{\jmath}2$ 中概述的直观过程来调度 $\tilde{\jmath}$，将其压缩到 $[m_1]$ 中任意可能数量的机器上，并以最小最大完工时间放置 $\tilde{\jmath}$。

**算法 2** 调度分配

**要求：** 备注6.1: $i_{min}=\arg\min_{i\in[m_1]}$ 负载$(i)+\frac{W_2}{i}$2: 压缩 $\tilde{\jmath}$，在 $i_{min}$ 台机器上尽早执行，机器编号为 1、.....、 $i_{min}$。

下面，我们将对上述算法（具体而言，即算法 2）给出一种新的、紧确的分析。注意，引理8和9 共同蕴含定理 2。因此，本节余下部分将致力于证明引理 9。我们将通过仔细分析 $S_1$ 中所有作业的总工作面积以及作业 $\tilde{\jmath}$在 $S_2$中的总工作面积来证明该引理。

**引理 9。** 设 $\lambda \geq \frac{2\sqrt{e}}{2\sqrt{e-1}}$。如果 $|S_2| = 1$，则算法 2 能够找到一个最大完工时间至多为 $\lambda$的连续单调可伸缩作业调度。

证明。 我们将 $S_2$ 中的单个作业记为 $\tilde{j}$。注意，$\gamma_{\tilde{j}}(\lambda - 1) > m_1$。否则，作业 $\tilde{j}$ 可以在机器 $1, \ldots, m_1$上执行，从而使最大完工时间至多为 $\lambda$ (参见图 1)。因此，以下我们假设 $\gamma_{\tilde{j}}(\lambda - 1) > m_1$。由于我们将 $W_2 := \sum_{j \in S_2} w_j(\gamma_j(\lambda - 1))$ 中的总工作量记为 $S_2$，因此有：

$$W_2 > (\lambda - 1)m_1. \tag{2}$$

结合引理 1.3 1a 和2c可得：

$$W_2 \leq m_1 - \frac{\lambda}{2}(m_1 - q) = (1 - \frac{\lambda}{2})m_1 + \frac{\lambda}{2}q. \tag{3}$$

最后，通过对方程（2）和（3）进行插值，我们得到存在一个 $c \in (0, 1]$ 满足：

$$W_2 = (1 - c)(\lambda - 1)m_1 + c(1 - \frac{\lambda}{2})m_1 + c\frac{\lambda}{2}q \tag{4}$$

命题 10。 未使用的机器数量 $q$ 有如下上界：

$$q \leq Q(c) := \frac{(1 - c)(\lambda - 1)m_1 + c(1 - \frac{\lambda}{2})m_1}{(1 - \frac{c}{2})\lambda}. \tag{5}$$

命题证明。 反设成立，则可得：

$$\begin{aligned}
& q > \frac{(1 - c)(\lambda - 1)m_1 + c(1 - \frac{\lambda}{2})m_1}{(1 - \frac{c}{2})\lambda} \\
\Leftrightarrow \quad & (1 - \frac{c}{2})\lambda q > (1 - c)(\lambda - 1)m_1 + c(1 - \frac{\lambda}{2})m_1 \\
\Leftrightarrow \quad & \lambda q > (1 - c)(\lambda - 1)m_1 + c(1 - \frac{\lambda}{2})m_1 + c\frac{\lambda}{2}q \\
\Leftrightarrow \quad & \lambda q > W_2. \qquad \text{使用（4）}
\end{aligned}$$

这与引理 1.3 3a直接矛盾。 ■

现在，我们定义临界机器数 $m' := \frac{2W_2}{\lambda}$，使得对于任意 $\hat{m} \geq m'$：

$$\frac{\lambda}{2} - \frac{W_2}{\hat{m}} \geq 0. \tag{6}$$

直观地说，该临界阈值的定义是：当 $\tilde{j}$ 被分配到多于 $m'$ 台机器时，其处理时间至多为 $\frac{\lambda}{2}$。接下来，我们注意到，引理 1.3 2b 中的处理时间至多为 $\frac{\lambda}{2}$ 的单个作业会导致一些需要单独考虑的边界情况。命题 11 给出了处理这些边界情况所需的条件，我们将在下面的分析中使用这些条件。

命题 11（边界情况）： 令 $\lambda \in [\frac{10}{7}, \frac{3}{2}]$则：

1. If $q + 1 < \lceil m' \rceil - 1$, then w.l.o.g. $\sum_{i=q+1}^{\lceil m' \rceil - 1} \text{load}(i) \geq \frac{\lambda}{2}(\lceil m' \rceil - q - 1)$ holds。
2. 如果 $q + 1 = \lceil m' \rceil - 1$，则算法 2 返回一个最大完工时间为最多 $\lambda$。

命题证明。 为提高可读性，我们将证明放在附录C中。 ■

为便于分析，我们定义一个函数 $f(q)$，该函数取决于实例和 $q$：

$$f(q) \coloneqq (\lambda-1)m_1 - \frac{\lambda}{2}q - W_2 \ln\left(\frac{m_1}{m'}\right). \tag{7}$$

注意，对于任意特定的固定实例，$W_2$ 和 $m'$ 都是常数，因此我们可以严格地将 $f$ 视为变量 $q$的函数进行分析。首先，我们将证明 $f(q) \geq 0$ 意味着算法2 返回的调度的最大完工时间至多为 $\lambda$。稍后，我们将在命题14中证明，对于所有满足 $f(q) \geq 0$ 的实例，$\lambda \in [\frac{10}{7}, \frac{3}{2}]$都成立。为准备证明，我们列出以下解析性质：

命题12（解析性质）。设 $\lambda \in [\frac{2\sqrt{e}}{2\sqrt{e}-1}, \frac{3}{2}]$，则：1. $\sum_{i=\lceil m'\rceil}^{m_1}\left[\frac{\lambda}{2} - \frac{W_2}{i}\right] > \int_{m'}^{m_1}\frac{\lambda}{2} - \frac{W_2}{x}\,dx$。2. $f(q) = (\lambda-1)m_1 - \frac{\lambda}{2}q - W_2\ln(\frac{m_1}{m'})$ 随 $q$ 单调递减。3. $Q(c) = (1-c)(\lambda-1)m_1 + c(1-\frac{\lambda}{2})m_1(1-\frac{c}{2})\lambda$ 随 $c$ 单调递增。4. $g(Q(c)) = (\lambda-1)m_1 - \frac{\lambda}{2}Q(c) - \lambda Q(c)\ln\frac{m_1\lambda}{2\lambda Q(c)}$ 随 $Q(c)$ 单调递增。

命题证明。 为提高可读性，我们将证明留到附录 D。 ■

命题 13。 令 $\lambda \in [\frac{10}{7}, \frac{3}{2}]$。若 $f(q) \geq 0$，则算法 2返回一个连续调度，其最大完工时间至多为 $\lambda$。

命题证明。 根据引理 1.3 1a，我们有 $S_1$ 的工作区域为：$W_1 \leq m_1 - W_2$。根据 $f$ (见方程 (7)) 的定义，我们有：

$$\begin{aligned}
& (\lambda-1)m_1 - \frac{\lambda}{2}q - W_2\ln(\frac{m_1}{m'}) \geq 0 \\
\Leftrightarrow\quad & (\lambda-1)m_1 - \frac{\lambda}{2}\frac{2}{\lambda}W_2 - \frac{\lambda}{2}q - W_2\ln(\frac{m_1}{m'}) \geq -W_2 \\
\Leftrightarrow\quad & \frac{\lambda}{2}(m_1-q) + \frac{\lambda}{2}(m_1 - \frac{2}{\lambda}W_2) - W_2\ln(\frac{m_1}{m'}) \geq m_1 - W_2 \\
\Rightarrow\quad & \frac{\lambda}{2}(m_1-q) + \frac{\lambda}{2}(m_1-m') - W_2\int_{m'}^{m_1}\frac{1}{x}\,dx \geq W_1 \\
\Leftrightarrow\quad & \frac{\lambda}{2}(m_1-q) + \int_{m'}^{m_1}\frac{\lambda}{2} - \frac{W_2}{x}\,dx \geq W_1.
\end{aligned} \tag{8}$$

结合命题 12.1，可得：

$$\frac{\lambda}{2}(m_1-q) + \sum_{i=\lceil m'\rceil}^{m_1}\left[\frac{\lambda}{2} - \frac{W_2}{i}\right] > W_1. \tag{9}$$

此外，由于 $S_1$ 中的作业按非递减顺序调度（见备注 6）：

$$W_1 = \sum_{i=1}^{m_1}\mathrm{load}(i) = \sum_{i=q+1}^{\lceil m'\rceil-1}\mathrm{load}(i) + \sum_{i=\lceil m'\rceil}^{m_1}\mathrm{load}(i). \tag{10}$$

**情况 1（$q+1 = \lceil m'\rceil - 1$）：** 在这种情况下，我们可以直接应用命题 11.2，得出算法2 返回的调度的最大完工时间至多为 $\lambda$。

**情况 2（$q+1<\lceil m'\rceil-1$）：** 在这种情况下，我们可以使用命题 11.1 和方程（10）来界定

$$W_1 \geq \frac{\lambda}{2}\left(\lceil m'\rceil - q - 1\right) + \sum_{i=\lceil m'\rceil}^{m_1} \text{load}(i). \tag{11}$$

合并方程（9）和（11）可得：

$$\begin{aligned}
& \frac{\lambda}{2}(m_1-q) + \sum_{i=\lceil m'\rceil}^{m_1}\left[\frac{\lambda}{2}-\frac{W_2}{i}\right] && > \frac{\lambda}{2}\left(\lceil m'\rceil - q - 1\right) + \sum_{i=\lceil m'\rceil}^{m_1}\text{load}(i) \\
\Leftrightarrow\ & \frac{\lambda}{2}(m_1-\lceil m'\rceil+1) + \sum_{i=\lceil m'\rceil}^{m_1}\left[\frac{\lambda}{2}-\frac{W_2}{i}\right] && > \sum_{i=\lceil m'\rceil}^{m_1}\text{load}(i) \\
\Leftrightarrow\ & \sum_{i=\lceil m'\rceil}^{m_1}\left[\lambda-\frac{W_2}{i}\right] && > \sum_{i=\lceil m'\rceil}^{m_1}\text{load}(i)
\end{aligned}$$

这意味着存在一个 $i'\in[\lceil m'\rceil, m_1]$，使得：

$$\lambda - \frac{W_2}{i'} > \text{load}(i') \Leftrightarrow \lambda > \text{load}(i') + \frac{W_2}{i'}.$$

请注意，$\frac{W_2}{i'}$对应于将单个作业 $\tilde{\jmath}\in S_2$ 从 $\gamma_{\tilde{\jmath}}$（$\lambda-1$）台机器压缩到 $i'$ 台机器时的最大处理时间。换句话说，由于单调性假设，我们可以在 $i'$ 台机器上处理单个作业 $\tilde{\jmath}\in S_2$，且最大完工时间不会超过 $\lambda$。请注意，机器 $\{1,\ \ldots,\ i'-1\}$ 的负载最多为 $\text{load}(i')$，因此该作业也可以安排在这些机器上。

**情况 3（$q+1>\lceil m'\rceil-1$）：** 在这种情况下，$\sum_{i=q+1}^{\lceil m'\rceil-1}\text{load}(i) = 0$ 成立。因此，方程（10）可简化为：

$$W_1 = \sum_{i=\lceil m'\rceil}^{m_1}\text{load}(i). \tag{12}$$

注意，$q$ 和 $\lceil m'\rceil$ 都是整数。因此，$q\geq\lceil m'\rceil-1$ 成立，或者等价地，$-\lceil m'\rceil+1\geq -q$。结合方程（9）和（12）可得：

$$\begin{aligned}
& \frac{\lambda}{2}(m_1-q) + \sum_{i=\lceil m'\rceil}^{m_1}\left[\frac{\lambda}{2}-\frac{W_2}{i}\right] && > \sum_{i=\lceil m'\rceil}^{m_1}\text{load}(i) \\
\Longrightarrow\ & \frac{\lambda}{2}(m_1-\lceil m'\rceil+1) + \sum_{i=\lceil m'\rceil}^{m_1}\left[\frac{\lambda}{2}-\frac{W_2}{i}\right] && > \sum_{i=\lceil m'\rceil}^{m_1}\text{load}(i) \\
\Leftrightarrow\ & \sum_{i=\lceil m'\rceil}^{m_1}\left[\lambda-\frac{W_2}{i}\right] && > \sum_{i=\lceil m'\rceil}^{m_1}\text{load}(i)
\end{aligned}$$

与情况 2 类似，这意味着存在一个 $i'\in[\lceil m'\rceil, m_1]$，使得：

$$\lambda - \frac{W_2}{i'} > \text{load}(i') \Leftrightarrow \lambda > \text{load}(i') + \frac{W_2}{i'}.$$

换言之，我们可以在 $\tilde{\jmath}$ 台 $i'$ 机器上处理作业，且最大完工时间不会超过 $\lambda$。至此，命题13的证明完成。■

我们现在已经证明，如果 $f(q)\geq 0$，那么算法 2 返回的调度的最大完工时间至多为$\lambda$。接下来需要证明，对于 $f(q)\geq 0$ 的所有有效取值，均有 $q$。

命题 14。 设 $\lambda \in [\frac{2\sqrt{e}}{2\sqrt{e}-1}, \frac{3}{2}]$。 则， $f(q) = (\lambda-1)m_1 - \frac{\lambda}{2}q - W_2 \ln(\frac{m_1}{m'}) \geq 0$。

命题证明。 注意，根据命题 12.2， $f(q)$ 关于 $q$ 单调递减。此外，根据命题 10，我们知道 $q \leq Q(c) = (1-c)(\lambda-1)m_1 + c(1-\frac{\lambda}{2})m_1(1-\frac{c}{2})\lambda$。因此，要证明 $f(q) \geq 0$，只需证明 $f(Q(c)) \geq 0$；在本证明的其余部分中，我们可以假设 $q = Q(c)$。

令 $Z := (1-c)(\lambda-1)m_1 + c(1-\frac{\lambda}{2})m_1$。根据方程（4），工作区域等于：

$$W_2 = Z + c\frac{\lambda}{2}Q(c).$$

根据定义： $Q(c) = \frac{Z}{(1-\frac{c}{2})\lambda} \Leftrightarrow Z = \lambda Q(c) - c\frac{\lambda}{2}Q(c)$。通过代换：

$$W_2 = \left(\lambda Q(c) - c\frac{\lambda}{2}Q(c)\right) + c\frac{\lambda}{2}Q(c) = \lambda Q(c). \tag{13}$$

使用方程（13）并代入 $m' = \frac{2W_2}{\lambda}$，我们可以将 $f(Q(c))$ 完全表示为 $Q(c)$的函数：

$$\begin{aligned} f(Q(c)) &= (\lambda-1)m_1 - \frac{\lambda}{2}Q(c) - W_2 \ln\frac{m_1}{m'} \\ &= (\lambda-1)m_1 - \frac{\lambda}{2}Q(c) - W_2 \ln\frac{m_1\lambda}{2W_2} \\ &= (\lambda-1)m_1 - \frac{\lambda}{2}Q(c) - \lambda Q(c)\ln\frac{m_1\lambda}{2\lambda Q(c)}. \qquad \text{using (13)} \end{aligned} \tag{14}$$

通过在这个最坏情况边界处对函数求值，之前保持不变的实例参数变为关于 $Q(c)$的函数依赖。令 $g(Q(c))$ 表示该求值表达式：

$$g(Q(c)) := (\lambda-1)m_1 - \frac{\lambda}{2}Q(c) - \lambda Q(c)\ln\left(\frac{m_1}{2Q(c)}\right). \tag{15}$$

根据命题 12.3， $Q(c)$ 关于 $c$单调递增。这意味着 $Q(c) \geq Q(0) = \frac{\lambda-1}{\lambda}m_1$。此外，根据命题 12.4， $g(Q(c))$ 关于 $Q(c)$ 单调递增。因此：

$$g(Q(c)) \geq g(Q(0)).$$

接下来只需证明 $g(Q(0)) \geq 0$。根据我们的初始假设，有 $\frac{2\sqrt{\epsilon}}{2\sqrt{\epsilon}-1} \leq \lambda$。我们可以重新整理该式，从而对常数 $\sqrt{e}$进行界定：

$$\begin{aligned} & \frac{2\sqrt{e}}{2\sqrt{e}-1} \leq \lambda \\ \Leftrightarrow \quad & 2\sqrt{e} \leq \lambda(2\sqrt{e}-1) \\ \Leftrightarrow \quad & 2\sqrt{e} \leq 2\lambda\sqrt{e} - \lambda \\ \Leftrightarrow \quad & \lambda \leq 2\sqrt{e}(\lambda-1) \\ \Leftrightarrow \quad & \frac{\lambda}{2(\lambda-1)} \leq \sqrt{e} \end{aligned}$$

由于 $\lambda \geq \frac{2\sqrt{e}}{2\sqrt{e}-1} > 1$，不等式两边均严格为正。因此，我们可以放心地取自然对数：

$$\begin{aligned} &\implies && \ln\left(\frac{\lambda}{2(\lambda-1)}\right) \leq \frac{1}{2} \\ &\Leftrightarrow && 0 \leq \frac{1}{2} - \ln\left(\frac{\lambda}{2(\lambda-1)}\right) \end{aligned}$$

接下来，我们对不等式进行缩放。由于 $m_1 \geq 1$ 和 $\lambda > 1$，项 $(\lambda-1)m_1$ 严格为正。两边乘以该项可保持不等式成立：

$$\begin{aligned} &\implies && 0 \leq (\lambda-1)m_1\left(\frac{1}{2} - \ln\left(\frac{\lambda}{2(\lambda-1)}\right)\right) \\ &\Leftrightarrow && 0 \leq \frac{1}{2}(\lambda-1)m_1 - (\lambda-1)m_1 \ln\left(\frac{\lambda}{2(\lambda-1)}\right) \\ &\Leftrightarrow && 0 \leq (\lambda-1)m_1 - \frac{1}{2}(\lambda-1)m_1 - (\lambda-1)m_1 \ln\left(\frac{\lambda}{2(\lambda-1)}\right) \end{aligned}$$

最后，我们代入 $Q(0) = \frac{\lambda-1}{\lambda} m_1$。注意到 $\lambda Q(0) = (\lambda-1)m_1$，我们可以重写系数并化简对数中的分式，从而重新构造函数 $g(Q(0))$：

$$\begin{aligned} &\Leftrightarrow && 0 \leq (\lambda-1)m_1 - \frac{\lambda}{2}Q(0) - \lambda Q(0)\ln\left(\frac{m_1}{2Q(0)}\right) \\ &\Leftrightarrow && 0 \leq g(Q(0)). && \text{using (15)} \end{aligned}$$

命题14的证明至此完成。 ■

命题 13 和命题 14 表明，算法 2 对于所有有效的 $\lambda$ 值以及 $c$ 和 $q$ 在 $|S_2| = 1$时，返回的调度的最大完工时间至多为。 □

引理 8 和 9证明了我们的主要结果。关于运行时间的证明，我们引用 [JO26], ，因为我们仅提供了新的分析，并未修改算法。

**定理 2**。 设 $\varepsilon > 0$。对于连续单调可塑作业调度，存在一个近似比为 $\frac{2\sqrt{e}}{2\sqrt{e}-1} + \varepsilon$ 且运行时间为 $O(nm$ 对数 $\frac{1}{\varepsilon})$的算法。

前面的分析表明，对于给定的面积 $\tilde{j}$，存在一个函数（见图 2），因此调度中的每个作业在相应位置上都必须具有更高的处理时间。我们在附录 B 中通过构造一个其总和低于该函数且任意接近该函数的实例，证明了这一分析是紧的。

## 3. 扩展到 CPU/GPU 调度

CPU/GPU 调度问题通过引入第二组$m_g$ 台相同的 GPU 机器，将此前讨论的 CPU 调度问题进行了推广。每个作业都可以作为可塑作业在 CPU 上调度，或依次在单台 GPU 机器上执行。

为了将我们改进后的 $\lambda = \frac{2\sqrt{e}}{2\sqrt{e}-1}$ 比率推广到 CPU/GPU 调度问题，我们采用双重近似框架，以及由 Germann 等人提出的多维多重选择背包问题 划分公式 [GJOT24]。我们将最大完工时间猜测记为 $d \in \mathbb{Q}_{>0}$。为了限制 GPU 上的调度，

Germann 等人 [GJOT24] 提出了一个大小函数 $\varphi(j)$，根据 GPU 作业的处理时间对其进行分类 $p(j)$：

$$\varphi(j) := \begin{cases} \max\{k \in \{0,1,2\} : \frac{kd}{3} \le p(j)\} & \text{if } p(j) \le d, \\ +\infty & \text{otherwise.} \end{cases} \tag{16}$$

通过限制分配给 GPU 分区（$\mathcal{T}_g$）的作业总量，我们可以保证，任一 GPU 机器上处理时间大于 $\frac{d}{3}$ 的作业最多只有两个。我们重述下面这个核心引理，它定义了将 GPU 最大完工时间与 CPU 最大完工时间匹配所需的松弛变量 $\delta$。

**引理 15** ([GJOT24])。设 $d \in \mathbb{Q}_{>0}$ 为最大完工时间猜测，$\delta \in \mathbb{Q}_{>0}$，且 $T_g \subseteq [n]$ 为分配给 *GPU* 的作业。若 $\sum_{j \in T_g} \varphi(j) \le 2m_g$，且 *GPU* 上的总工作量满足 $\sum_{j \in T_g} p(j) \le m_g(1+\delta)d$，则我们可以在 $O(n \log n)$ 时间内，使用最长处理时间优先（*LPT*）调度构造一个长度至多为 $(\frac{4}{3}+\delta)d$ 的 *GPU* 调度。

要将此结果应用于我们的新结果，需要使 GPU 调度与我们的 CPU 比率 $\lambda$匹配。因此，我们将松弛常数 $\delta$ 设定为使 GPU 最大完工时间等于 CPU 最大完工时间：

$$\frac{4}{3} + \delta = \lambda \implies \delta = \frac{2\sqrt{e}}{2\sqrt{e}-1} - \frac{4}{3} \approx 0.1018 \tag{17}$$

定义了 $\delta$ 后，我们可以通过求解多维多重选择背包问题，找到一个有效的作业划分，将作业分配到 CPU 层架（$S_0$，$S_1$，$S_2$）和 GPU（$\mathcal{T}_g$）上。为确保动态规划在多项式时间内运行，GPU 的处理时间将按缩放因子 $\mu := \frac{n}{\delta d m_g}$进行缩放和舍入，具体为 $\tilde{p}(j) := \lfloor \mu p(j) \rfloor$。

**引理 16**（背包问题表述）。通过动态规划，可以在时间 $O(n^2 m_c m_g)$内找到满足引理 *5* 中所确立的 *CPU* 约束，以及 *GPU* 约束 $\sum_{j \in T_g} \varphi(j) \le 2m_g$ 和 $\sum_{j \in T_g} \tilde{p}(j) \le \frac{n}{\delta}$ 的划分。

证明。该结论可由多维多重选择背包问题MMCKP的标准动态规划表述 [KPP得出。动态规划方法的细节与 [GJOT中所用的方法类似。

□

**定理 4**。设 $\varepsilon > 0$。存在一种用于 *CPU/GPU* 调度的 $(\frac{2\sqrt{e}}{2\sqrt{e}-1} + \varepsilon)$-近似算法，其运行时间为 $O(\log(nm_c m_g/\varepsilon) \cdot n^2 m_c m_g)$。

证明。利用引理 16 中的框架，我们可以在 $O(n^2 m_c m_g)$ 时间内计算出跨 CPU 和 GPU 的有效作业划分。对于 CPU 划分，第 4 节中的算法保证调度长度至多为 $\lambda d = \frac{2\sqrt{e}}{2\sqrt{e}-1} d$。对于 GPU 划分，背包约束确保 $\sum_{j \in T_g} \varphi(j) \le 2m_g$，且忽放宽工作负载被限制为 $m_g(1+\delta)d$。根据引理 15，LPT 算法将在时间$O(n \log n)$内构造出长度至多为 $(\frac{4}{3}+\delta)d = \lambda d$ 的有效 GPU 调度。

采用原始近似框架，我们可以使用二分搜索过程，迭代 $O(\log(nm_c m_g/\varepsilon))$次。由于每次迭代的运行时间主要由求解 MMCKP 决定，因此总时间复杂度上界为 $O(\log(nm_c m_g/\varepsilon) \cdot n^2 m_c m_g)$。

□

## 4. 结论

本文对连续单调可塑作业调度（CMMJS）问题进行了改进分析。通过细致的几何论证，我们成功地将近似比从约 $1.459+\varepsilon$ [JO26] 降至 $\frac{2\sqrt{e}}{2\sqrt{e}-1}+\varepsilon \approx 1.435+\varepsilon$。此外，我们通过给出一个匹配的下界实例，证明该算法方法的最坏情况分析基本上是紧的。我们还展示了该算法的通用性，将其无缝扩展到异构 CPU/GPU 调度问题，并获得相同的近似保证。

尽管取得了这些改进，仍有几个自然的开放问题有待解决。未来研究的一个主要理论重点，是确定这一设定下具有实际效率的算法的真正下界。由于我们的划分所具有的结构性质（引理 7）对任意 $\lambda \in [\frac{10}{7}, \frac{3}{2}]$ 都成立，因此，算法 2 是否能够得到增强——或者是否可以开发出一种新的分配策略——从而将近似比降至 $\frac{10}{7} \approx 1.428$ 屏障以下，仍然是一个引人关注的开放问题。

最后，另一个重要方向是进一步改善运行时间。虽然我们的算法具有实际效率，其时间复杂度为 $O(nm \log \frac{1}{\varepsilon})$，但如果能降低对机器数量 $m$的依赖，使其类似于 [GJO23； JLL16], ，则该算法将在 $n$上实现近似线性的运行时间，而我们目前的运行时间在 $n$ 上是二次的（如果 $m \approx n$）。

## 参考文献

[BKM+15] R. Bleuse、S. Kedad-Sidhoum、F. Monna、G. Mounié 和 D. Trystram。“使用 GPU 加速器在多核上调度独立任务”。载于：*Concurrency and Computation: Practice and Experience* 27.6 (2015)，第 1625–1638 页。doi: 10.1002/CPE.3359.[DL89] J. Du 和 J. Y. Leung。“并行任务系统的调度复杂度”。载于：*SIAM* 离散数学杂志 2.4 (1989)，第 473–487 页。doi: 10.1137/0402042.[DLM06] T. Decker、T. Lücking 和 B. Monien。“用于调度同类可塑任务的 54 近似算法”。载于：理论计算机科学 361.2-3 (2006)，第 226–240 页。doi: 10.1016/j.tcs.2006.05.012.[Dro95] M. Drozdowski。“多处理器任务调度的复杂度”。载于：《波兰科学院技术科学通报》43 (1995)，第 381–392 页。[GJO23] K. Grage、K. Jansen 和 F. Ohnesorge。“利用压缩和卷积改进单调可塑作业调度算法”。载于：*Euro-Par 2023*：并行处理——第 *29* 届并行与分布式计算国际会议。计算机科学讲义。2023。doi: 10.1007/978-3-031-39698-4_34.[GJOT24] S. B. Germann、K. Jansen、F. Ohnesorge 和 M. Tutas。“CPU/GPU 调度的 3/2 对偶近似”。载于：第 *22* 届国际实验算法研讨会，*SEA 2024*。2024。doi: 10.4230/LIPICS.SEA.2024.13.[Gra69] R. L. Graham。“多处理时序异常的界”。载于：*SIAM* 应用数学杂志 17.2 (1969)，第 416–429 页。[HS87] D. S. Hochbaum 和 D. B. Shmoys。“将对偶近似算法用于调度问题：理论与实践结果”。载于：《*ACM* 杂志》34.1 (1987)，第 144–162 页。doi: 10.1145/7531.7535.

[Jan10] K. Jansen。“同类处理器上的作业调度 EPTAS：使用含有常数个整数变量的混合整数线性规划松弛”。载于：SIAM 离散数学杂志 24.2 (2010)，第 457–485 页。doi: 10.1137/090749451。网址： https://doi.org/10.1137/090749451.[JGJ78] E. G. C. Jr.、M. R. Garey 和 D. S. Johnson。“装箱在多处理器调度中的应用”。载于：SIAM 计算杂志 7.1 (1978)，第 1–17 页。doi: 10.1137/0207001.[JL18] K. Jansen 和 F. Land。“线性时间调度单调可塑作业”。载于：2018 IEEE 国际并行与分布式处理研讨会（IPDPS 2018），温哥华，加拿大不列颠哥伦比亚省，2018 年 5 月 21–25 日。IEEE 计算机学会，2018，第 172–181 页。doi: 10.1109/IPDPS.2018.00027.[JLL16] K. Jansen、F. Land 和 K. Land。“调度与装箱问题算法运行时间的界”。载于：SIAM 离散数学杂志 30.1 (2016)，第 343–366 页。doi: 10.1137/140952636.[JO26] K. Jansen 和 F. Ohnesorge。“连续单调可塑作业调度的实用 73/50 近似”。载于：第 43 届国际计算机科学理论方面研讨会，STACS 2026。LIPIcs。Schloss Dagstuhl - Leibniz 信息学中心，2026。doi: 10.4230/LIPICS.STACS.2026.56.[Joh06] B. Johannes。“最小化最大完工时间的并行作业调度”。载于：J. Sched. 9.5 (2006)，第 433–452 页。doi: 10.1007/S10951-006-8497-6。网址： https://doi.org/10.1007/s10951-006-8497-6.[JP99] K. Jansen 和 L. Porkolab。“可塑并行任务调度的线性时间近似方案”。载于：第十届 ACM-SIAM 离散算法年度研讨会论文集（SODA 1999）。ACM/SIAM，1999，第 490–498 页。网址： http://dl.acm.org/citation.cfm?id=314500.314870.[JR19] K. Jansen 和 M. Rau。“弥合伪多项式条带装箱的间隙”。载于：第 27 届欧洲算法研讨会，ESA 2019。LIPIcs。Schloss Dagstuhl - Leibniz 信息学中心，2019，62:1–62:14。doi: 10.4230/LIPICS.ESA.2019.62。网址：https://doi.org/10.4230/LIPIcs.ESA.2019.62.[JT08] K. Jansen 和 R. Thöle。“并行作业调度的近似算法：突破 2 的近似比”。载于：自动机、语言与程序设计，第 35 届国际研讨会，ICALP 2008。计算机科学讲义。Springer，2008，第 234–245 页。doi: 10.1007/978-3-540-70575-8_20.[KPP04] H. Kellerer、U. Pferschy 和 D. Pisinger。背包问题。Springer，2004。ISBN：978-3-540-40286-2。doi: 10.1007/978-3-540-24777-7.[LT94] W. Ludwig 和 P. Tiwari。“可塑与不可塑并行任务的调度”。载于：第五届 ACM-SIAM 离散算法年度研讨会论文集（SODA 1994）。ACM/SIAM，1994，第 167–176 页。网址：http://dl.acm.org/citation.cfm?id=314464.314491.[MRT07] G. Mounie、C. Rapine 和 D. Trystram。“独立单调可塑任务的 3/2 近似算法”。载于：SIAM 计算杂志 37.2 (2007)，第 401–412 页。doi: 10.1137/S0097539701385995。

[TWY92] J. Turek、J. L. Wolf 和 P. S. Yu。“可并行任务调度的近似算法”。载于：第四届 *ACM* 并行算法与体系结构研讨会论文集（*SPAA 1992*）。1992 年，第 323–332 页。DOI： 10.1145/140901.141909。

[WZC23] F. Wu、X. Zhang 和 B. Chen。“单调可变任务调度的改进近似算法”。载于：《欧洲运筹学杂志》306.2（2023），第 567–578 页。doi：10.1016/J.EJOR.2022.08.034。

## A. $S_2$ 中至多有一个作业

本节将证明以下引理。

**引理 8**。设 $\lambda \geq \frac{2\sqrt{e}}{2\sqrt{e}-1}$。如果 $|S_2| > 1$，则我们可以构造一个最大完工时间至多为 $\lambda$ 的连续调度。

为了证明这个引理，我们对 $q$的取值进行分类讨论。如果 $q$ 较大，我们可以直接利用面积约束证明 $S_2$ 中至多有一个作业。如果 $q$ 较小，我们可以通过算法3找到一个可行调度。如前所述，此过程与 [JO26]中使用的过程相同。

**观察 17**。令 $\lambda \geq \frac{2\sqrt{e}}{2\sqrt{e}-1}$。若 $q > \frac{1}{3}(1 - \frac{1}{\sqrt{e}})m_1$，则 $|S_2| \leq 1$。

证明。反设，即 $|S_2| > 1$。由引理 1.3 1a、2c 和3a，可得：

$$
\begin{aligned}
m_1 &\overset{1a}{\geq} \sum_{j \in S_1} w_j(\gamma_j(1)) + W_2 \\
&> \frac{\lambda}{2}(m_1 - q) + 2\lambda && (|S_2| > 1，\text{引理 1.3 2c 和3a}) \\
&= \frac{\lambda}{2}m_1 + \frac{3}{2}\lambda q \\
&= \frac{\lambda}{2}m_1 + \frac{3}{2}\lambda\left(\frac{1}{3}\left(1 - \frac{1}{\sqrt{e}}\right)m_1\right) \\
&= \frac{\lambda}{2}m_1 + \frac{\lambda}{2}\left(1 - \frac{1}{\sqrt{e}}\right)m_1 \\
&= \frac{\lambda}{2}m_1\left(2 - \frac{1}{\sqrt{e}}\right) \\
&= \lambda m_1\left(\frac{2\sqrt{e}-1}{2\sqrt{e}}\right).
\end{aligned}
$$

比较首末两项并除以 $m_1$，得到：

$$1 > \lambda\left(\frac{2\sqrt{e}-1}{2\sqrt{e}}\right) \implies \lambda < \frac{2\sqrt{e}}{2\sqrt{e}-1}.$$

这就产生了矛盾。 □

现在我们证明，如果 $q$ 足够小，就可以通过算法3找到一个可行调度。

**算法 3** RepairS2

1: **当** $m_2 > m_1$ **时执行**2: 令 $j$ 为 $S_2$中处理时间最短的作业。3: 令 $p$ 为 $j$ 分配到的机器数量。4: 将 $j$ 分配给 $p-1$ 台机器。5: **结束循环**6: 按执行时间降序排列 $S_1$ 中的作业。7: 按执行时间升序排列 $S_2$ 中的作业。

**观察 18**。 设 $\lambda \geq \frac{2\sqrt{e}}{2\sqrt{e}-1}$。如果 $q \leq \frac{1}{3}(1-\frac{1}{\sqrt{e}})m_1$，那么我们可以通过算法 *3* 找到一个可行调度。

证明。 我们通过分析算法 3 来证明这一陈述。以下将 $S_1$ 中作业的总工作面积记为 $W_1$：$= \sum_{j\in S_1} w_j(\alpha_j)$，并将 $S_2$ 中作业的总工作面积记为 $W_2$：$= \sum_{j\in S_2} w_j(\alpha_j)$。
命题 19。 $W_1 > \frac{1+2\sqrt{e}}{6\sqrt{e}-3}m_1$和 $W_2 < \frac{4\sqrt{e}-4}{6\sqrt{e}-3}m_1$。

命题证明。 根据引理 1.32c 我们有：

$$W_1 > \frac{\lambda}{2}(m_1 - q) \geq \frac{1}{2}\frac{2\sqrt{e}}{2\sqrt{e}-1}(m_1 - \frac{1}{3}(1-\frac{1}{\sqrt{e}})m_1) = \frac{1+2\sqrt{e}}{6\sqrt{e}-3}m_1$$

并且 $W_1 + W_2 \leq m_1$意味着 $W_2 \leq m_1 - W_1 < m_1 - \frac{1+2\sqrt{e}}{6\sqrt{e}-3}m_1 = \frac{4\sqrt{e}-4}{6\sqrt{e}-3}m_1$。 ■

命题 20： 算法 3 执行后， $S_2$ 中的每个作业都至少在 3 台机器上执行。

命题证明。 在应用算法 3之前，每个作业 $j \in S_2$ 都在 $\gamma_j$（$\lambda - 1$）台机器上执行。根据 $\gamma_j$的定义，每个作业的处理时间至多为 $\lambda - 1 = \frac{1}{2\sqrt{e}-1}$。此外，根据引理 1.3 3a， $S_2$ 中的每个作业的工作区域至少为 $\lambda q \geq \frac{2\sqrt{e}}{2\sqrt{e}-1}$。这两个性质意味着， $S_2$ 中的每个作业初始时都被分配到至少 $\frac{\lambda q}{\lambda - 1} \geq 2\sqrt{e} > 3$ 台机器上。

为导出矛盾，假设在算法 3执行期间的某个时刻，作业 $j$ 从 3 台机器压缩到 2 台机器上。由于算法 3始终选择处理时间最短的作业进行压缩，这意味着在这一确切步骤， $S_2$ 中的每个剩余作业当前的处理时间至少为 $t_j(3)$。

我们可以利用间隙 $q$ 以及作业的单调性，为 $t_j(3)$建立一个严格的下界。回想一下，由于工作必须高于 $\lambda q$ 且机器分配必须高于 $q$ 才能安排到 $S_2$中，因此 $S_2$ 中的作业无法以至多 $\lambda$ 的处理时间安排在 $q$ 台机器的间隙中。因此，根据单调性，必须有 $t_j(q) > \lambda$。根据 $q$的大小，我们区分两种情况：

**情况 1**（$q \geq 3$）**：** 我们应用处理时间函数 $t_j$ 是非递增的这一单调性。这直接得到 $t_j(3) \geq t_j(q) > \lambda > \frac{\lambda}{3}$。

**情况 2**（$q < 3$）**：** 由于 $q$ 是一个整数， $q \in \{1, 2\}$。我们应用工作区域函数 $w_j$ 是非递减的这一单调性。这给出 $w_j(3) \geq w_j(q) = q \cdot t_j(q) > q\lambda \geq \lambda$。因此，在 3 台机器上的处理时间为 $t_j(3) = \frac{w_j(3)}{3} > \frac{\lambda}{3}$。

在这两种情况下，我们都有 $t_j(3) > \frac{\lambda}{3}$。因此，在进行此次压缩时， $S_2$ 中的每个作业的处理时间都严格大于 $\frac{\lambda}{3}$。

令 $m_2$ 表示当前分配给 $S_2$中作业的机器总数。由于算法WHILE中的循环仍在运行，我们知道 $m_2 > m_1$。 $S_2$ 的当前总工作面积严格大于 $\frac{\lambda}{3}m_2 > \frac{\lambda}{3}m_1$。

由于作业的工作面积随机器数量增加而非递减，因此将作业压缩到更少的机器上不会增加其总工作面积。因此，初始总工作面积 $W_2$ 必定不小于当前总工作面积，即 $W_2 > \frac{\lambda}{3} m_1$。由于 $\lambda \geq \frac{2\sqrt{e}}{2\sqrt{e}-1}$，我们有：

$$W_2 > \frac{2\sqrt{e}}{3(2\sqrt{e}-1)} m_1 = \frac{2\sqrt{e}}{6\sqrt{e}-3} m_1.$$

由于 $e < 4$，因此 $\sqrt{e} < 2$，这意味着 $2\sqrt{e} > 4\sqrt{e} - 4$。由此，我们得到：

$$W_2 > \frac{4\sqrt{e}-4}{6\sqrt{e}-3} m_1.$$

这与命题 19严格矛盾，该命题给出的面积上界为 $W_2 < \frac{4\sqrt{e}-4}{6\sqrt{e}-3} m_1$。由此可知，该算法不会将任何作业压缩到 2 台机器上。■

命题 21。 在算法 3 执行之前， $S_2$ 中的每个作业的处理时间都大于 $\frac{4}{10\sqrt{e}-5}$。

命题证明。 我们根据机器数量进行分类讨论 $\gamma_j(\lambda - 1)$，即作业 $j$ 在算法 3 执行之前所运行的机器数量：

**情况 1（$\gamma_j(\lambda-1) \leq 4$）：** 每个作业 $j$ 的面积至少为 $\lambda$，根据引理 1.3 3a。这意味着处理时间至少为 $\frac{\lambda}{4} > \frac{4}{10\sqrt{e}-5}$。

**情况 2（$\gamma_j(\lambda-1) > 4$）：** 在这种情况下，我们得到：

$$h_j(\gamma_j(\lambda-1)) > (\lambda-1)\frac{\gamma_j(\lambda-1)-1}{\gamma_j(\lambda-1)} \geq (\lambda-1)\frac{4}{5} = \frac{4}{10\sqrt{e}-5}.$$

第一个不等式根据 $\gamma_j$ 的定义成立，即 $\gamma_j(x) = \min\{\alpha \in [m] \mid h_j(\alpha) \leq x\}$。 ■

命题 22。 在算法 3之后， $S_2$ 中的每个作业的处理时间至多为 $\frac{4}{3}\frac{4\sqrt{e}-4}{6\sqrt{e}-3}$。

命题证明。 为导出矛盾，假设 $S_2$ 中至少存在一个作业，其处理时间严格大于 $\frac{4}{3}\frac{4\sqrt{e}-4}{6\sqrt{e}-3}$。

回顾一下，算法 3 会迭代地压缩当前处理时间最短的作业。由于该算法始终选择处理时间最短的作业进行压缩，如果当前最小处理时间超过此阈值，那么当前 $S_2$ 中的所有作业的处理时间都必须至少为 $\frac{4}{3}\frac{4\sqrt{e}-4}{6\sqrt{e}-3}$。

这意味着总工作面积 $W_2$ 满足：

$$W_2 \geq \left(\frac{4\sqrt{e}-4}{6\sqrt{e}-3}\right) \cdot m_1.$$

这与 $W_2$ 在命题19中确立的上界。 ■

现在，我们已经证明了证明引理9所需的所有性质。为导出矛盾，假设算法 3 返回了一个无效调度。设 $i$ 是总负载为 $\text{load}(i) > \lambda \geq \frac{2\sqrt{e}}{2\sqrt{e}-1}$的机器。根据构造，这台机器恰好执行两个作业，一个来自 $S_1$，另一个来自 $S_2$。我们将该机器上来自 $S_2$ 的作业的处理时间记为 $t$。由于该作业至少被压缩过一次，因此有 $t \in (\lambda - 1, \frac{4}{3}\frac{4\sqrt{e}-4}{6\sqrt{e}-3})$。

现在考虑任意一台机器 $i' \in \{i+1, \ldots, m_1\}$。由于 $S_1$ 中的作业按处理时间非递减顺序排列，因此机器 $i'$ 上的来自 $S_1$ 的作业，其处理时间至少为 $\lambda - t$。根据命题 20，我们知道来自 $S_2$ 的作业，其处理时间至少为 $\frac{4}{10\sqrt{e}-5}$。因此：

$$\text{load}(i') \geq \lambda - t + \frac{4}{10\sqrt{e}-5} > \frac{\sqrt{e}+1}{2\sqrt{e}-1} \qquad \forall i' \in \{i+1, \ldots, m_1\} \tag{18}$$

我们对 $i$ 左侧的机器采用类似的论证。

考虑任意一台机器 $i' \in \{q+2、\ldots\ldots、i-1\}$：由于 $S_2$ 中的作业按处理时间非递增排序，因此来自 $S_2$ 的作业在机器 $i'$ 上的处理时间至少为 $t \geq \lambda - 1$。根据引理 1.3 2b，$S_1$ 中的每个作业（机器 $q+1$ 上的作业除外）的处理时间至少为 $\frac{\lambda}{2}$。因此：

$$\text{load}(i') \geq \frac{\lambda}{2} + t \geq \frac{\lambda}{2} + (\lambda - 1) = \frac{3\lambda}{2} - 1 \geq \frac{\sqrt{e}+1}{2\sqrt{e}-1} \qquad \forall i' \in \{q+2, \ldots, i-1\} \tag{19}$$

现在考虑遗漏的机器 $q+1$ 和 $i$。由于机器 $i$ 上的负载至少为 $\lambda$，这两台机器的总负载至少为：

$$\begin{aligned}
\lambda + (\lambda - 1) &\geq 2\frac{2\sqrt{e}}{2\sqrt{e}-1} - 1 \\
&= 2\frac{2\sqrt{e}}{2\sqrt{e}-1} - \frac{2\sqrt{e}-1}{2\sqrt{e}-1} \\
&= \frac{2\sqrt{e}+1}{2\sqrt{e}-1} \\
&= \frac{2\sqrt{e}+2}{2\sqrt{e}-1} - \frac{1}{2\sqrt{e}-1} \\
&= 2\frac{\sqrt{e}+1}{2\sqrt{e}-1} - \frac{1}{2\sqrt{e}-1}.
\end{aligned}$$

将此结果与方程（18）和（19）结合，可知从 $q+1$ 到 $m_1$ 的所有机器的总面积至少为 $\frac{\sqrt{e}+1}{2\sqrt{e}-1}(m_1 - q) - \frac{1}{2\sqrt{e}-1}$。由于剩余的 $q$ 台机器的负载至少为 $\lambda - 1 = \frac{1}{2\sqrt{e}-1}$，因此总面积至少为：

$$\frac{\sqrt{e}+1}{2\sqrt{e}-1}(m_1 - q) - \frac{1}{2\sqrt{e}-1} + \frac{1}{2\sqrt{e}-1}q.$$

由于所有作业的总面积至多为 $m_1$ 根据引理 1.31a，我们有：

$$\begin{aligned}
&\frac{\sqrt{e}+1}{2\sqrt{e}-1}(m_1 - q) - \frac{1}{2\sqrt{e}-1} + \frac{1}{2\sqrt{e}-1}q && < m_1 \\
\Longrightarrow\ &\frac{\sqrt{e}+1}{2\sqrt{e}-1}m_1 - \frac{\sqrt{e}}{2\sqrt{e}-1}q - \frac{1}{2\sqrt{e}-1} && < m_1 \\
\Longrightarrow\ &\frac{\sqrt{e}+1}{2\sqrt{e}-1}m_1 - \frac{2\sqrt{e}-1}{2\sqrt{e}-1}m_1 - \frac{1}{2\sqrt{e}-1} && < \frac{\sqrt{e}}{2\sqrt{e}-1}q \\
\Longrightarrow\ &\frac{2-\sqrt{e}}{\sqrt{e}}m_1 - \frac{1}{\sqrt{e}} && < q \\
\Longrightarrow\ &(\frac{2-\sqrt{e}}{\sqrt{e}} - \frac{1}{\sqrt{e}m_1})m_1 && < q
\end{aligned}$$

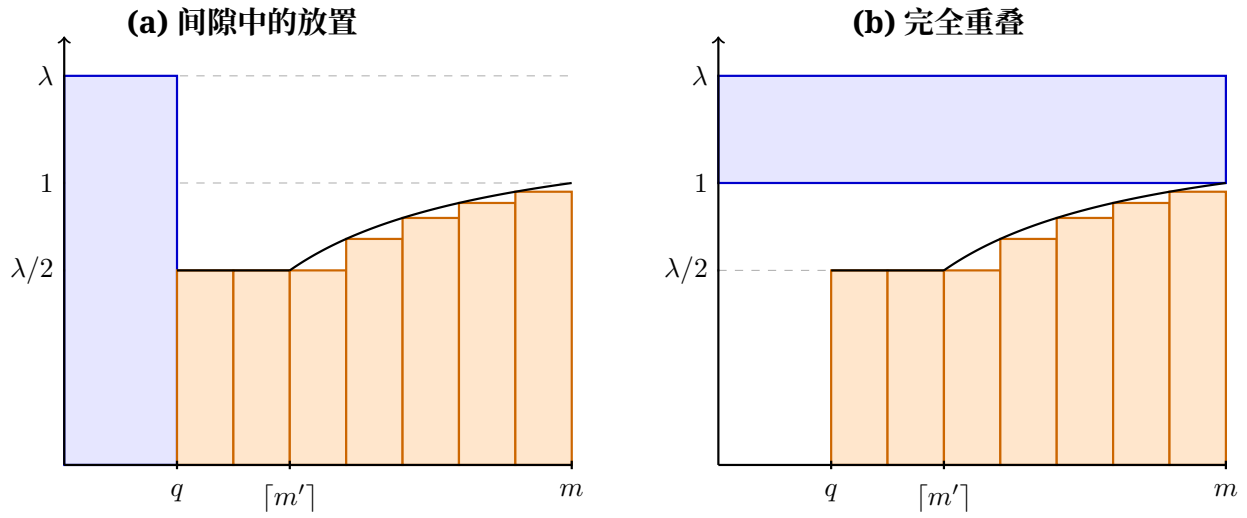



图 2：小作业以橙色显示，大作业以蓝色表示。

注意到 $q \geq 1$ 以及条件 $q < \frac{1}{3}(1 - \frac{1}{\sqrt{e}})m_1 < 0.132m_1$ 由观察18可推出 $m_1 > 7$。对于 $m_1 > 7$，可得 $\frac{2-\sqrt{e}}{\sqrt{e}} - \frac{1}{\sqrt{e}m_1} > 0.137$，从而将不等式化简为 $q > 0.137m_1$。这直接导致矛盾。

□

观察 17和观察18 证明了引理 9

## B. 算法 2的最坏情况实例

在本节中，我们给出一个实例，使算法2的近似比任意接近 $\frac{2\sqrt{e}}{2\sqrt{e}-1}$ 从而表明我们的分析是紧的。

**引理 3。** 对于任意 $\varepsilon > 0$，存在一个实例 $I$，使得算法 *1* 返回的调度的最大完工时间为 $\mathrm{ALG}(I)$，且 $\mathrm{ALG(I)}\ \mathrm{OPT}(I) \geq \frac{2\sqrt{e}}{2\sqrt{e}-1} - \varepsilon$。

证明。 设 $\lambda = \frac{2\sqrt{e}}{2\sqrt{e}-1}$。我们构造一个由 $\varepsilon$ 参数化的实例族。我们将展示，当机器数量 $m$ 足够大并按后文所述选取时，某个 $\varepsilon' \leq \varepsilon$ 的比率。令 $q = \frac{1}{2\sqrt{e}}m$ 表示取整后某个特定的机器索引。我们定义一个恰好包含 $m - \lfloor q \rfloor + 1$ 个作业的实例。一个大作业 $\tilde{j}$，其面积恰好为 $W_2 = (\lambda - 1)m$，以及 $m - \lfloor q \rfloor$ 个小作业，其索引为 $j \in \{\lfloor q \rfloor + 1, \ldots, m\}$。

我们定义临界机器阈值 $m' = \frac{2W_2}{\lambda} = \frac{2}{(\lambda-1)\lambda}m$。我们仔细定义小作业的处理时间，使得如果将大作业压缩并放置在它们上方，则最右侧机器的负载恰好为 $\lambda$（减去一个小误差）。

我们为每个小作业定义常数工作面积函数：

$$w_j(\cdot) := \begin{cases} \frac{\lambda}{2} & \text{if } j \in (q, \lceil m' \rceil) \\ \lambda - \frac{W_2}{j} - \delta & \text{if } j \in [\lceil m' \rceil, m] \end{cases} \tag{20}$$

其中 $\delta = \frac{\varepsilon' + \frac{\lambda}{2}(q - \lfloor q \rfloor)}{m - \lceil m' \rceil + 1}$ 是一个小误差，用于确保小作业的总面积恰好为 $m - W_2$，这将在下一个命题中说明。因此，整个

实例恰好为 $W_2 + (m - W_2) = m$。因此，可以通过为每个作业分配 1 台 $m$机器并连续执行这些作业，得到最大完工时间为 *1* 的最优调度。命题 *23*。由方程（20）定义的作业的总工作面积为 $m - W_2$。

命题证明。方程 (20) 的总工作面积为:

$$\begin{aligned} W_1 \coloneqq \sum_{j=\lfloor q \rfloor+1}^{m} w_j(1) &= \frac{\lambda}{2}(m - \lfloor q \rfloor) + (m - \lceil m' \rceil + 1)(\frac{\lambda}{2} - \delta) - \sum_{j=\lceil m' \rceil}^{m} \frac{W_2}{j} \\ &= \frac{\lambda}{2}(m - \lfloor q \rfloor) - (m - \lceil m' \rceil + 1)\delta + \sum_{j=\lceil m' \rceil}^{m} \left[\frac{\lambda}{2} - \frac{W_2}{j}\right] \\ &= \frac{\lambda}{2}(m - \lfloor q \rfloor) - (\varepsilon' + \frac{\lambda}{2}(q - \lfloor q \rfloor)) + \sum_{j=\lceil m' \rceil}^{m} \left[\frac{\lambda}{2} - \frac{W_2}{j}\right] \\ &= \frac{\lambda}{2}(m - q) - \varepsilon' + \sum_{j=\lceil m' \rceil}^{m} \left[\frac{\lambda}{2} - \frac{W_2}{j}\right]. \end{aligned} \tag{21}$$

由于 $\sum_{i=\lceil m' \rceil}^{m} \left[\frac{\lambda}{2} - \frac{W_2}{i}\right]$ 更大且收敛于 $\int_{m'}^{m} \frac{\lambda}{2} - \frac{W_2}{x}\, dx$（命题 12.1），我们可以找到一个 $\varepsilon' \le 2\varepsilon$, 并选择足够大的 $m$，使得 $\frac{\lambda}{2}\,\frac{(q - \lfloor q \rfloor)}{m - \lceil m' \rceil + 1} \le \frac{\varepsilon}{2}$，并且

$$\sum_{i=\lceil m' \rceil}^{m} \left[\frac{\lambda}{2} - \frac{W_2}{i}\right] = \int_{m'}^{m} \frac{\lambda}{2} - \frac{W_2}{x}\, dx + \varepsilon'. \tag{22}$$

现在利用方程 (21) 和 (22)，我们得到

$$W_1 = \frac{\lambda}{2}(m - q) - \varepsilon' + \int_{m'}^{m} \frac{\lambda}{2} - \frac{W_2}{x}\, dx + \varepsilon'. \tag{23}$$

此外，我们有

$$\begin{aligned} (\lambda - 1)m - \frac{\lambda}{2}q - W_2 \ln(\frac{m}{m'}) &= (\lambda - 1)m - \frac{\lambda}{2}(\frac{1}{2\sqrt{e}}m) - (\lambda - 1)\ln\left(\frac{m}{\frac{2(\lambda-1)}{\lambda}m}\right) \\ &= (\lambda - 1)m - \frac{1}{2}(\lambda - 1)m - \frac{1}{2}(\lambda - 1) = 0. \end{aligned} \tag{24}$$

方程 (24) 现在可推出，类似于方程 (8) 中的结论

$$\begin{aligned} & (\lambda - 1)m - \frac{\lambda}{2}q - W_2 \ln(\frac{m}{m'}) = 0 \\ \Leftrightarrow\quad & (\lambda - 1)m - \frac{\lambda}{2}\frac{2}{\lambda}W_2 - \frac{\lambda}{2}q - W_2 \ln(\frac{m}{m'}) = -W_2 \\ \Leftrightarrow\quad & \frac{\lambda}{2}(m - q) + \frac{\lambda}{2}(m - \frac{2}{\lambda}W_2) - W_2 \ln(\frac{m}{m'}) = m - W_2 \\ \Leftrightarrow\quad & \frac{\lambda}{2}(m - q) + \frac{\lambda}{2}(m - m') - W_2 \int_{m'}^{m} \frac{1}{x}\, dx = m - W_2 \\ \Leftrightarrow\quad & \frac{\lambda}{2}(m - q) + \int_{m'}^{m} \frac{\lambda}{2} - \frac{W_2}{x}\, dx = m - W_2. \end{aligned} \tag{25}$$

结合方程（23）和（25）可得

$$W_1 = m - W_2. \qquad \blacksquare$$

为了证明所述近似比，我们现在必须证明该算法返回的调度的最大完工时间为 $\lambda = \frac{2\sqrt{e}}{2\sqrt{e}-1}$。

首先，注意到由于工作函数是常数，任意划分 $C_1 \sqcup C_2 \sqcup C_3$ 都满足引理 7中的性质。因此，算法 1 将小作业放置在 $S_0 \sqcup S_1$ 并将唯一的大作业 $\tilde{j}$ 放置在 $S_2$。随后，算法会按照处理时间非递减的顺序，将小作业分别分配到机器 $\lfloor q \rfloor + 1$、……、$m$中的每台机器上。然后，由于 $|S_2| = 1$ 算法 2 会尝试将大作业 $\tilde{j}$ 放置在机器 1、……、$i$ 上，其中 $i \in [m]$。我们考察 $i$所有可能选择下的最大完工时间，图 2展示了两个特殊情况：

**情况 1（间隙中的放置）：** 如果将 $\tilde{j}$ 压缩到恰好 $i = \lfloor q \rfloor$ 台机器上，它就能放入空隙中，而不会与任何小作业重叠。其处理时间为：

$$\frac{W_2}{\lfloor q \rfloor} = \frac{(\lambda-1)m}{\lfloor \frac{1}{2\sqrt{e}} m \rfloor} \geq \frac{(\lambda-1)m}{\frac{1}{2\sqrt{e}} m} = 2\sqrt{e}(\lambda-1) = 2\sqrt{e}\left(\frac{2\sqrt{e}}{2\sqrt{e}-1} - \frac{2\sqrt{e}-1}{2\sqrt{e}-1}\right) = \lambda.$$

**情况 2（部分重叠）：** 若 $\tilde{j}$ 被压缩到$(q, \lceil m' \rceil)$ 台机器上，则会与第一块小作业重叠。机器 $i$ 上的总负载等于小作业的处理时间加上大作业在 $i$台机器上的处理时间。根据方程（6）中 $m'$ 的定义，我们得到：

$$\frac{\lambda}{2} + \frac{W_2}{i} \geq \frac{\lambda}{2} + \frac{W_2}{m'} > \frac{\lambda}{2} + \frac{\lambda}{2} = \lambda.$$

**情况 3（完全重叠）：** 若 $\tilde{j}$ 被压缩到 $[\lceil m' \rceil, m]$ 台机器上，则会与第二块小作业重叠。机器 $i$ 上的总负载再次等于小作业的处理时间加上大作业的处理时间：

$$t_j(i) + \frac{W_2}{i} = \left(\lambda - \frac{W_2}{i} - \delta\right) + \frac{W_2}{i} = \lambda - \delta.$$

由于 $\delta = \frac{\varepsilon' + \frac{\lambda}{2} - \left(\frac{q - \lfloor q \rfloor}{m - \lceil m' \rceil + 1}\right)}{\leq \frac{-\frac{\varepsilon}{2} + \varepsilon}{m - \lceil m' \rceil + 1} \leq \varepsilon}$，因此得到的最大完工时间至少为 $\lambda - \varepsilon$。 □

## C. 边界情况

本节证明以下命题：

命题 11（边界情况）。 令 $\lambda \in [\frac{10}{7}, \frac{3}{2}]$则：

1. If $q + 1 < \lceil m' \rceil - 1$, then w.l.o.g. $\sum_{i=q+1}^{\lceil m' \rceil - 1} \text{load}(i) \geq \frac{\lambda}{2}(\lceil m' \rceil - q - 1)$ holds
。2. 如果 $q + 1 = \lceil m' \rceil - 1$，则算法 2 返回一个最大完工时间为

most $\lambda$.

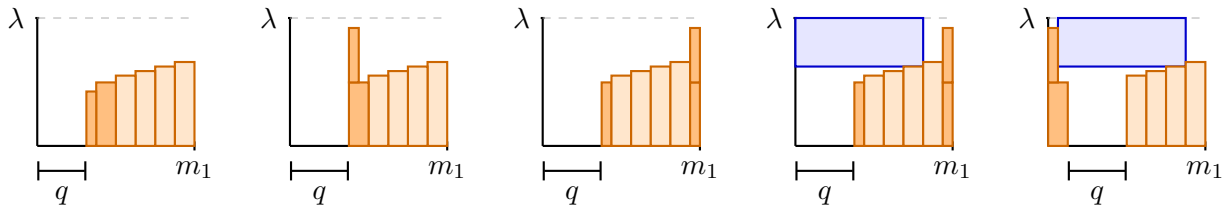
我们将通过证明每个单独的命题来完成这一点。注意， $q + 1 < \lceil m' \rceil$ 意味着 $q + 2 \leq \lceil m' \rceil - 1$，因为 $q$ 是一个整数，因此在下文中我们将在这一条件下证明该命题。

**观察 24。** 如果 $q + 2 \leq \lceil m' \rceil - 1$，那么我们可以对备注 *6* 给出的调度进行转换，使其同时满足以下性质：

$$\sum_{i=q+1}^{\lceil m' \rceil - 1} \text{load}(i) \geq \frac{\lambda}{2}(\lceil m' \rceil - q - 1)$$

证明。 在这种情况下，我们假设作业 $j^1$ 来自引理 1.3 2b ，且其处理时间至多为 $\frac{\lambda}{2}$ 存在。如果该作业不存在， $S_1$ 中的任何作业的处理时间都至少为 $\frac{\lambda}{2}$ ，这直接意味着 $\sum_{i=q+1}^{\lceil m' \rceil - 1}$ 负载$(i) \geq \frac{\lambda}{2}($ $\lceil m' \rceil - q - 1)$。注意，选择情况 $q+2 \leq \lceil m' \rceil - 1$ 是为了确保总和至少包含两项，从而保证该总和包含索引$q+1$ 和 $q+2$。

情况 1（负载$(q+1)$ + 负载$(q+2)$ $\leq \lambda$）： 由于 $j^1$ 是处理时间最短的作业，且调度中的负载按递增顺序排列，因此将 $j^1$ 调度到机器 $q+1$上。我们可以将 $j^1$ 重新调度到机器 $q+2$上。注意，这会改变 $q$。由于合并后的负载至多为 $\lambda$，因此最大完工时间不会增加超过 $\lambda$。随后，我们将调整调度，并通过将机器 $q+2$ 排序到正确位置，使其变为非连续的，从而使负载再次保持非递减顺序。由于此时调度具有与之前相同的性质（连续性除外），这简化了后续分析。如果负载$(q+1)$ +负载$(q+2)$ $> 1$，我们将 $m_1$ 减 1，因为此时该机器属于 $S_0$的一部分。使用算法 2 将作业放置在 $S_2$ 中后，我们可以通过将分离的作业向左移动，使该调度再次变为连续的。这一变换如图 3 所示。现在，通过将步骤 (a)-(c) 作为预处理，并将步骤 (e) 作为后处理，对算法进行扩展。由于在步骤 (c) 之后， $q+1$ 和 $m_1$ 之间的每台机器的负载至少为 $\frac{\lambda}{2}$，因此该结论成立。




(a) 初始状态。 (b) 重新调度 $j^*$。 (c) 重新排序。 (d) 算法 2。 (e) 连续。

图 3：将低高度的单个作业重新调度的转换。

情况 2（负载$(q+1)$ +负载$(q+2)$ $> \lambda$）： 在这种情况下，我们可以将总和界定为 $\sum_{q+1}^{\lceil m' \rceil - 1}$ 负载$(i)$ =负载$(q+1)$ +负载$(q+2)$ + $\sum_{i=q+3}^{\lceil m' \rceil - 1}$ 负载$(i)$> $\lambda + \frac{\lambda}{2}((\lceil (m' \rceil - 1) - (q+3) + 1) = \frac{\lambda}{2}($ $\lceil m' \rceil - q - 1)$，其上界为 2b。 □

**观察 25**。 令 $\lambda \in [\frac{10}{7}, \frac{3}{2}]$和 $q+1 = \lceil m' \rceil - 1$，则算法 2的比率为 $\lambda$。

证明。 我们可以假设 $W_2 > \lambda q$ ，因为否则 $W_2 \leq \lambda q$ 将成立，这意味着我们可以将 $\tilde{j}$ 压缩到 $q$ 台机器上，且处理时间至多为 $\lambda$。该条件等价于 $\frac{W_2}{\lambda} > q$。根据我们的初始假设，这意味着 $\frac{W_2}{\lambda} + 1 > q + 1 = \lceil m' \rceil - 1 \geq m' - 1 = \frac{2W_2}{\lambda} - 1$。这可以转化为

$$2 \geq \frac{W_2}{\lambda} \Leftrightarrow 2\lambda \geq W_2 \tag{26}$$

这意味着我们可以将 $\tilde{j}$ 压缩到 2 台机器上，且处理时间至多为 $\lambda$。如果 $q \geq 2$成立，那么我们就这样做即可。于是剩余的情况是 $q = 1$ （因为情况 $q = 0$ 的比率已在 [JO26]中得到证明）。为了证明该比率对所有 $m_1$都成立，我们将陈述一些性质；如果算法 2 生成的调度的最大完工时间大于 $\lambda$ ，则这些性质必须成立，并证明

这些性质彼此矛盾。定义 $W_1 := \sum_{j \in S_1} w_j(\gamma_j(1))$。

$$\text{负载}(i) + \frac{W_2}{i} > \lambda \quad \text{对于所有 } i \in \{2, \ldots, m_1\} \quad \text{因为 } j \text{ 并未被压缩到 } i \text{ 台机器上。} \tag{27}$$

$$W_2 > \lambda \quad \text{by Lemma 1.3 3a} \tag{28}$$

$$W_1 + W_2 \le m_1 \quad \text{by Lemma 1.3 1a.} \tag{29}$$

**情况 1 ($m_1 = 2$)：** 我们有：

$$W_1 + W_2 = \text{load}(2) + W_2 \overset{Equation\ (27)}{>} \lambda - \frac{W_2}{2} + W_2 = \lambda + \frac{W_2}{2} \overset{Equation\ (28)}{>} \frac{3}{2}\lambda > 2 = m_1$$

这与方程 (29) 矛盾。

**情况 2 ($m_1 = 3$)：** 这也会导致矛盾：$W_1 + W_2 = \text{负载}(2) + \text{负载}(3) + W_2 \overset{Equation\ (27)\ )(28)}{>} \lambda - \frac{W_2}{2} + \lambda - \frac{W_2}{3} + W_2 = 2\lambda + \frac{1}{6}W_2 > 3 = m_1$。

**情况 3 ($m_1 = 4$)：** 在这种情况下，我们通过以下方式得到矛盾：$W_1 + W_2 = \text{负载}(2) + \text{负载}(3) + \text{负载}(4) + W_2 > 3\lambda - \frac{W_2}{2} - \frac{W_2}{3} - \frac{W_2}{4} + W_2 = 3\lambda - \frac{1}{12}W_2 \overset{Equation\ (26)}{>} 3\lambda - \frac{2}{12}\lambda > 4 = m_1$。

**情形4（$m_1 \ge 5$）：** 在这一剩余情形中，我们可以利用以下事实：$W_2$ 压缩到 $m_1$ 台机器上的处理时间必须大于 $\lambda - 1$，否则我们就可以将其放在那里并完成任务。这将同样导致矛盾，表明我们需要的面积超过了最多可用的面积1a。

$$\begin{aligned} W_1 + W_2 &\overset{2c}{>} \frac{\lambda}{2}(m_1 - 1) + W_2 & &\overset{2}{>} \frac{\lambda}{2}(m_1 - 1) + (\lambda - 1)m_1 \\ &= \frac{\lambda}{2}m_1 - \frac{\lambda}{2} + \lambda m_1 - m_1 & &= \lambda(\frac{3}{2}m_1 - \frac{1}{2}) - m_1 \\ &\ge \frac{10}{7}(\frac{3}{2}m_1 - \frac{1}{2}) - m_1 & &= \frac{8}{7}m_1 - \frac{5}{7} \\ &= m_1 + (\frac{1}{7}m_1 - \frac{5}{7}) & &\overset{m_1 \ge 5}{\ge} m_1 \end{aligned}$$

再次与方程（29）矛盾。 □

观察24 和25现证明命题11。

## D. 分析证明

本节将证明以下命题：

命题12（解析性质）。设 $\lambda \in [\frac{2\sqrt{e}}{2\sqrt{e}-1}, \frac{3}{2}]$，则：

1. $\sum_{i=\lceil m' \rceil}^{m_1} \left[\frac{\lambda}{2} - \frac{W_2}{i}\right] > \int_{m'}^{m_1} \frac{\lambda}{2} - \frac{W_2}{x}\, dx$。 2. $f(q) = (\lambda - 1)m_1 - \frac{\lambda}{2}q - W_2 \ln(\frac{m_1}{m'})$ 随 $q$ 单调递减。 3. $Q(c) = (1-c)(\lambda-1)m_1 + c(1-\frac{\lambda}{2})m_1(1-\frac{c}{2})\lambda$ 随 $c$ 单调递增。 4. $g(Q(c)) = (\lambda - 1)m_1 - \frac{\lambda}{2}Q(c) - \lambda Q(c) \ln \frac{m_1\lambda}{2\lambda Q(c)}$ 随 $Q(c)$ 单调递增。

下面我们将通过分别证明每个命题来完成归档。

命题 2.3.1。 $\sum_{i=\lceil m' \rceil}^{m_1} \left[ \frac{\lambda}{2} - \frac{W_2}{i} \right] > \int_{m'}^{m_1} \frac{\lambda}{2} - \frac{W_2}{x} \, dx$

命题证明。定义 $h(i) := \frac{\lambda}{2} - \frac{W_2}{i}$ 以及 $k := \lceil m' \rceil$。

注意，根据方程（6），$h(i)$ 成立；对于每个 $i \in [1, m_1]$，以下结论成立，因为$h(i)$是严格单调的：

$$\int_{i-1}^{i} h(x) \, dx < \int_{i-1}^{i} h(i) \, dx = h(i)$$

这意味着：

$$\int_{k}^{m_1} h(x) \, dx = \sum_{i=k+1}^{m_1} \int_{i-1}^{i} h(x) \, dx \leq \sum_{i=k+1}^{m_1} h(i)$$

对于取整部分，同样有：

$$\int_{m'}^{k} h(x) \, dx \leq \int_{m'}^{k} h(k) \, dx = (k - m')h(k) < h(k)$$

综合可得：

$$\int_{m'}^{m_1} h(x) \, dx = \int_{m'}^{k} h(x) \, dx + \int_{k}^{m_1} h(x) \, dx < h(k) + \sum_{i=k+1}^{m_1} h(i) = \sum_{i=\lceil m' \rceil}^{m_1} h(i). \qquad \blacksquare$$

命题2.3.2。如果 $\lambda \in [\frac{4}{3}, \frac{3}{2}]$ 且 $q \leq Q(c)$，那么 $f(q) = (\lambda - 1) \ln \binom{m_1}{m'}$ 关于 $q$ 单调递减。

命题证明。考虑 $f$ 关于 $q$ 的一阶导数，我们有：

$$\frac{\partial f}{\partial q} = -\frac{\lambda}{2}.$$

给定有界区间 $\lambda \in [\frac{4}{3}, \frac{3}{2}]$，显然有 $-\frac{\lambda}{2} \leq -\frac{2}{3} < 0$。 $\blacksquare$

命题 2.3.3。 如果 $\lambda \in [1, \frac{3}{2}]$，则 $Q(c) = \frac{(1-c)(\lambda-1)m_1 + c(1=\frac{\lambda}{2})m_1(1-\frac{c}{2})\lambda}{}$ 关于 $c$ 单调递增。

命题证明。 我们将再次计算 $Q(c)$ 的导数。提出常数 $\frac{m_1}{\lambda}$ 并化简分子 $u(c)$：

$$\begin{aligned} u(c) &= (1 - c)(\lambda - 1) + c\left(1 - \frac{\lambda}{2}\right) \\ &= \lambda - 1 + 2c - \frac{3}{2}c\lambda \end{aligned}$$

将 $v(c)$ 定义为分母并计算导数：

$$\begin{aligned} u'(c) &= 2 - \frac{3}{2}\lambda \\ v(c) &= 1 - \frac{c}{2} \implies v'(c) = -\frac{1}{2} \end{aligned}$$

应用商法则求新的分子 $u'v - uv'$:

$$\begin{aligned} u'v - uv' &= \left(2 - \frac{3}{2}\lambda\right)\left(1 - \frac{c}{2}\right) - \left(\lambda - 1 + 2c - \frac{3}{2}c\lambda\right)\left(-\frac{1}{2}\right) \\ &= \left(2 - c - \frac{3}{2}\lambda + \frac{3}{4}c\lambda\right) + \left(\frac{1}{2}\lambda - \frac{1}{2} + c - \frac{3}{4}c\lambda\right) \\ &= \frac{3}{2} - \lambda = \frac{3 - 2\lambda}{2}. \end{aligned}$$

将所有部分与因式分解后的常数 $\frac{m_1}{\lambda}$ 合并，得到最终导数:

$$\begin{aligned} \frac{\partial Q(c)}{\partial c} &= \frac{m_1}{\lambda} \cdot \frac{\frac{3-2\lambda}{2}}{\left(1 - \frac{c}{2}\right)^2} \\ &= \frac{m_1(3 - 2\lambda)}{2\lambda\left(1 - \frac{c}{2}\right)^2}. \end{aligned}$$

现在我们得到:

$$\frac{\partial Q(c)}{\partial c} = \frac{m_1(3-2\lambda)}{2\lambda(1-\frac{c}{2})^2} \overset{!}{\geq} 0 \Leftrightarrow 3 - 2\lambda \geq 0 \Leftrightarrow \frac{3}{2} \geq \lambda. \qquad \blacksquare$$

命题 2.3.4. 如果 $\lambda \geq \frac{2\sqrt{e}}{2\sqrt{e}-1}$，那么 $g(Q(c)) = (\lambda - 1)m_1 - \frac{\lambda}{2}Q(c) - \lambda Q(c) \ln \frac{m_1\lambda}{2\lambda Q(c)}$ 关于 $Q(c)$ 单调递增。

命题证明。 首先，我们计算 $g$ 的导数。在对数内约去 $\lambda$ 以简化表达式:

$$\ln\left(\frac{m_1\lambda}{2\lambda Q(c)}\right) = \ln\left(\frac{m_1}{2Q(c)}\right)$$

逐项对 $Q(c)$求导:

$$\begin{aligned} \frac{\partial g}{\partial Q(c)} &= \frac{\partial}{\partial Q(c)}\left[(\lambda - 1)m_1 - \frac{\lambda}{2}Q(c) - \lambda Q(c)\ln\left(\frac{m_1}{2Q(c)}\right)\right] \\ &= 0 - \frac{\lambda}{2} - \frac{\partial}{\partial Q(c)}\left[\lambda Q(c)\ln\left(\frac{m_1}{2Q(c)}\right)\right] \end{aligned}$$

对最后一项应用乘积法则 $(uv)' = u'v + uv'$:

$$\begin{aligned} \frac{\partial}{\partial Q(c)}\left[\lambda Q(c)\ln\left(\frac{m_1}{2Q(c)}\right)\right] &= \lambda\ln\left(\frac{m_1}{2Q(c)}\right) + \lambda Q(c)\left(-\frac{1}{Q(c)}\right) \\ &= \lambda\ln\left(\frac{m_1}{2Q(c)}\right) - \lambda \end{aligned}$$

回代，合并常数并提取 $\frac{\lambda}{2}$:

$$\begin{aligned} \frac{\partial g}{\partial Q(c)} &= -\frac{\lambda}{2} - \left(\lambda\ln\left(\frac{m_1}{2Q(c)}\right) - \lambda\right) \\ &= \frac{\lambda}{2} - \lambda\ln\left(\frac{m_1}{2Q(c)}\right) \\ &= \frac{\lambda}{2}\left(1 - 2\ln\left(\frac{m_1}{2Q(c)}\right)\right) \end{aligned}$$

我们现在得到:

$$\begin{aligned}\frac{\partial g(c)}{\partial Q(c)} =& \frac{\lambda}{2}\left(1-2\ln\frac{m_1}{2Q(c)}\right)\overset{!}{\geq}0\\ \Leftrightarrow\quad & \frac{1}{2}\geq\ln\frac{m_1}{2Q(c)}\\ \Leftrightarrow\quad & \sqrt{e}\geq\frac{m_1}{2Q(c)}\\ \Leftrightarrow\quad & Q(c)\geq\frac{m_1}{2\sqrt{e}}\end{aligned}$$

And:

$$\begin{aligned}&\lambda\geq\frac{2\sqrt{e}}{2\sqrt{e}-1}=\frac{1}{1-\frac{1}{2\sqrt{e}}}\\ \Leftrightarrow&\lambda(1-\frac{1}{2\sqrt{e}})\geq 1\\ \Leftrightarrow&\lambda-1\geq\lambda\frac{1}{2\sqrt{e}}\\ \Leftrightarrow&\frac{\lambda-1}{\lambda}m_1\geq\frac{m_1}{2\sqrt{e}}\end{aligned}$$

■

## E. 完整算法

在本 节中，我们为完整性和清晰起见，重新阐述完整算法，来自 [JO26]。 .

划分 $C_1 \sqcup C_2 \sqcup C_3 = J_B$ 通过求解多重选择背包问题得到，具体形式见引理5。该算法的总体概述见算法1。具体而言，算法1中的第2行和第3行由算法4实现。

**Alg算法 4** 分配 g算法 分配 g划分

要求：如引理5所述的划分 $C_1, C_2, C_3$ $\lambda \in [\frac{2\sqrt{e}}{2\sqrt{e}-1}, \frac{3}{2}]$ 以及最大完工时间猜测 $d = 1$。1：对于每个 $j \in C_1 \sqcup C_2$，设置 $\alpha_j \leftarrow \gamma_j(\lambda)$；若 $t_j(\alpha_j) > 1$，则将 $j$ 移至 $S_0$，否则移至 $S_1$。2：按照处理时间非递减的顺序，将 $S_0 \sqcup S_1$ 调度到 $\sum_{j \in S_0 \sqcup S_1} \alpha_j$ 台机器上。3：▷可能使用多于 $m$台机器。4：当存在不同的机器 $i_1, i_2$，它们仅分配了满足 $\alpha_j = 1$ 的作业，且load($i_1$) ≤load($i_2$) ≤ $\frac{\lambda}{2}$ 时，执行以下操作：5：将所有作业从 $i_2$ 移至 $i_1$。6：结束循环。7：如果存在一台机器 $i$，其仅分配了满足 $\alpha_j = 1$ 的作业，且load($i$) ≤ $\frac{\lambda}{2}$ 并且load($i$) +load($m_1$) ≤ $\lambda$，则执行以下操作：8：将所有作业从 $i$ 移至 $m_1$。9：结束条件判断。10：当存在不同的机器三元组{$i_1, i_1+1, i_1+2$}和{$i_2, i_2+1, i_2+2$}，它们仅分配了满足 $\alpha_j = 3$ 的作业，且load($i_1$) ≤ load($i_2$) ≤ $\frac{\lambda}{2}$ 时，执行以下操作：11：将所有作业从{$i_2, i_2+1, i_2+2$}移至{$i_1, i_1+1, i_1+2$}。12：结束循环。▷此时，最多使用 $m$ 台机器。13：记 $q := m - \sum_{i, load(i)>0} 1$ 为未使用机器的数量。14：对于每个作业 $j \in C_3$，执行以下操作：15：设置 $\alpha_j \leftarrow \gamma_j(\lambda - 1)$。16：如果$w_j(\alpha_j) \le \lambda q$ 或 $\alpha_j \le q$，则：17：将 $j$ 加入 $C_1 \sqcup C_2$，并重新执行前述步骤（第1–13行）。18：否则：19：将 $j$ 加入 $S_2$。20：结束条件判断。21：结束循环。22：移除未使用的机器，使机器总数仅剩 $m$ 台。23：按负载非递减顺序对机器进行（连续）排序。