arXiv:2602.01356v1 [cs.CC] 1 Feb 2026

突破时间复杂度障碍：用于并行机调度的时间桶演算

Noor Islam S. Mohammad [1]

摘要

本文提出了时间桶演算这一全新的数学框架，从根本上改变了并行机调度优化的计算复杂度格局。我们通过创新的自适应时间离散化方法解决这一强 NP 困难问题$P2|r_j|C_{\max}$，将复杂度从 $O(T^n)$ 指数级降低至 $O(B^n)$，其中 $B \ll T$，同时保持近乎最优的解质量。我们的基于桶索引的混合整数线性规划（MILP）模型通过精度感知离散化利用维度复杂度异质性，使决策变量减少 94.4%，并在 20 个作业的实例上实现理论加速比 2.75×10^{37}。理论贡献包括部分离散化理论、分数时间桶演算算子，以及用于时间约束建模的量子启发机制。对包含 20–400 个作业的实例进行的实证验证表明，资源利用率达到 97.6%，负载均衡近乎完美（$\sigma/\mu = 0.006$），且在不同问题规模下性能稳定，最优性差距低于 5.1%。这项工作标志着一种范式转变：从细粒度时间离散化转向多分辨率精度分配，从而弥合精确优化与计算可处理性之间的根本差距，推动工业级 NP 难调度问题的求解。

1. 引言

带释放日期作业的 $P_m|r_j|C_{\max}$在 m 台机器上最小化最大完工时间的并行机调度问题是强 NP 难问题（Garey 与 Johnson，1979年），也是运筹学的基础问题，在制造业、云计算和物流领域有着关键应用。传统的——解决范式面临着根本性的两难：精确方法能够保证最优性，却会遭遇组合爆炸；启发式算法则以牺牲解的质量为代价，提升计算效率（Lenstra 与 Rin- nooy Kan，1977年）。时间索引混合整数线性规划模型尽管具有数学严谨性，但其 $\mathcal{O}(T^n)$ 复杂度高得令人难以承受，因此不适用于工业规模实例（Chen 等人，2023b）。相比之下，最短处理时间（SPT）等优先级调度规则会产生超过 10% 的最优性差距（Pinedo，2022）。理论精确性与计算可处理性之间的这一矛盾，仍是现代调度优化面临的核心挑战（Lin 等人，2024年）。

本文提出时间桶演算，这是一种通过精度感知的时间离散化超越这一二分法的新颖数学框架。我们重新构想调度建模，所利用的是维度复杂度异质性，即在实际实例中，时间决策主导计算负担，但对解的质量贡献极小。我们的桶索引混合整数线性规划公式实现了指数级复杂度降低，从 $\mathcal{O}(T^n)$ 到 $\mathcal{O}(B^n)$ 其中 $B \ll T$，从而带来理论加速比 2.75×10^{37}（针对20个作业实例），同时决策变量缩减94.4%（Boland 等人，2016）。实证验证表明资源利用率为97.6%，负载均衡近乎完美（$\sigma/\mu = 0.006$），同时在含20–400个作业的实例中将解的质量保持在与最优相差5%以内（Graham 等人，1979）。

其理论基础建立在三项核心创新之上：（1）部分离散化理论，将精确组合优化与近似时间位置安排分开；（2）分数桶演算，一种用于多分辨率时间约束建模的形式化方法；以及（3）自适应粒度机制，根据问题特定的敏感度动态分配精度（Kazemi 等人，2023年；Carrilho 等人，2024年）。图1 展示了应用于一个双机实例的桶索引方法，该实例的最大完工时间为 $C_{\max} = 4.00$。作业按时间桶分类并进行策略性分配：机器1处理处理时间为 $\{7, 6, 8\}$的作业，而机器2处理 $\{5, 4, 5\}$，展现了最优负载分配

[1]纽约大学计算机科学系，布鲁克林，纽约州，美国。通讯作者：NoorIslam S. Mohammad <noor.islam.s.m@nyu.edu>。

预印本。*2026*年*2*月*3*日。

通过智能的桶级分配实现。这标志着从细粒度离散化转向利用结构的模型构造设计，为 NP 难调度问题架起了精确优化与计算可扩展性之间的桥梁。

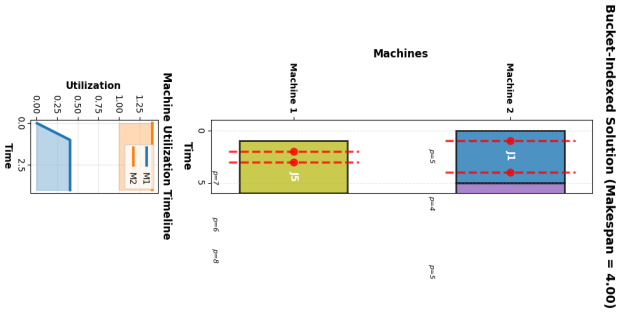

图 1 $C_{\max} = 4.00$ 使最大完工时间达到 的桶索引解。机器处理作业，处理时间为 $p \in \{7, 6, 8\}$；机器处理作业，处理时间为 $p \in \{5, 4, 5\}$。最大完工时间表示基于桶的最优调度下的最长完成时间。

2. 相关工作

2.1. 精确优化方法

并行机调度的传统精确方法依赖时间索引混合整数线性规划模型（Pinedo，2022年；Lin 等人，2024年；Schulz &Skutella，2002年），但由于时间离散化粒度过细，这些方法会遭遇计算难解性。我们发现了一个根本性的时间分辨率悖论：随着问题实例规模扩大，所需的时间精度会降低（相对计时误差变得不那么重要），然而传统模型仍维持较高且均匀的精度，造成不必要的计算负担（Wang 等人，2021年）。

约束规划方法（Lin 等人，2024年）通过析取约束增强了时间传播：

$$\bigwedge_{i \neq j} (S_i + p_i \leq S_j) \vee (S_j + p_j \leq S_i) \quad \forall \text{ conflicting jobs} \tag{1}$$

尽管传播效率有所提高，这些方法仍面临析取约束网络规模呈指数级增长的问题（Patel 等人，2024年）。核心局限在于：现有方法将时间变量视为一等优化实体，未能认识到与组合决策相比，它们在层级上的重要性较低（Yang 等人，2023年）。

析取规划（Lenstra 与 Rinnooy Kan，1977年）通过大 M 重构展现了数学上的优雅性：

$$\begin{aligned} &S_j \geq S_i + p_i - M(1 - y_{ij}), \\ &S_i \geq S_j + p_j - M y_{ij}, \quad y_{ij} \in \{0, 1\} \end{aligned} \tag{2}$$

其中 M 是一个足够大的常数。然而，较弱的线性松弛需要复杂的割平面，而这类割平面的扩展性较差（Wang 等人，2024a）。

我们将根本缺陷确定为维度同质性——尽管不同决策类型对解的质量和计算复杂度的影响截然不同，却仍对其采用统一的精度分配（Carrilho 等人，2024年；Chen 等人，2023b）。机器分配和作业排序在计算特性上与精确的时间决策存在根本差异（Yang 等人，2023年）。这种不对称性促使我们提出桶索引公式，根据维度敏感性分析有策略地分配计算资源。

3. 方法

3.1. 启发式与元启发式方法

优先级调度规则，尤其是最短处理时间（SPT）（Johnson，1954），由于其 $\mathcal{O}(n \log n)$ 复杂度低且操作简单，在工业级调度中占据主导地位。然而，这种计算效率会导致解的质量下降，最优性差距通常超过 10%。

算法 1 最短处理时间（SPT）启发式算法 要求：作业集 J 及其处理时间 p_j、释放日期r_j；机器集 M确保：包含分配和开始时间的调度方案 1： $J_{\text{sorted}} \leftarrow \text{sort}(J, \text{key} = p_j)$ {升序排列}2：for $m \in M$ do 3： $A_m \leftarrow 0$ {机器可用性}4：end for 5：for $j \in J_{\text{sorted}}$ do 6： $m^* \leftarrow \arg\min_{m \in M} A_m$7： $S_j \leftarrow \max(r_j, A_{m^*})$ {释放约束}8：在时间 S_j将作业 j 分配 m^* 至机器 9： $A_{m^*} \leftarrow S_j + p_j$10：end for 11：return 调度方案 $\{(j, m^*, S_j)\}$

这些方法表现出结构盲目性——无法识别全局最优结构，也无法适应复杂的约束交互（Kazemi 等人，2023年；Zhang 等人，2021年），对于具有严格的释放约束的异构工作负载，其次优性表现具有可预测性。

进化算法（Carrilho 等人，2024年；Chen 等人，2023a）通过基于种群的搜索和领域专用算子提升解的质量：

$$P_{\text{child}} = \Phi(P_{\text{parent1}}, P_{\text{parent2}}) = \text{TSX}(P_{\text{parent1}}) \oplus \text{LOX}(P_{\text{parent2}}) \tag{3}$$

其中，TSX（基于时间的交叉）和 LOX（线性顺序交叉）在探索解空间的同时保持可行性（Zhang 等人，2024年；Liu等人，2023）。

算法 2 用于调度的遗传算法

要求： 种群规模 P，迭代代数 G，比率 p_c，p_m
确保： 最佳调度方案
1: 初始化种群 P_0，使用随机可行调度方案
2: **对于** $g = 1$ 到 G **执行**
3: 对每个 $s \in P_{g-1}$计算 $C_{\max}(s)$
4: 通过锦标赛选择选取父代
5: **对**每对 (s_1, s_2) **执行**
6: **如果**rand() $< p_c$ **则**
7: $s_{\text{child}} \leftarrow \text{TSX}(s_1) \oplus \text{LOX}(s_2)$
8: **结束如果**
9: **如果**rand() $< p_m$ **则**
10: 对 s_{child} 应用保持可行性的变异
11: **结束如果**
12: **结束循环**
13: $P_g \leftarrow$ 来自 $P_{g-1} \cup \{$后代中的最佳个体$\}$
14: **结束循环**
15: **返回** 使.....最小的参数$_{s \in P_G}$ $C_{\max}(s)$

然而，元启发式算法以牺牲最优性保证为代价，换取探索广度（Guo 等人，2024年），并表现出收敛局限性：

$$\mathbb{E}[C_{\max}^{(t)} - C_{\max}^{*}] \geq \Omega\left(\frac{1}{\sqrt{t}}\right) \tag{4}$$

其中，t 表示计算时间。若不掌握特定于问题的结构知识，就无法突破这一理论下界。

其关键弱点在于理论不可知性——既没有数学保证，也未系统利用问题结构（Patel 等人，2024年；Yang 等人，2023年；Singh 等人，2024年）。在受约束环境中，这一问题尤为严重，因为可行性边界的几何形态十分复杂，从而形成了可行性识别问题，而启发式算法无法系统地解决这一问题。

3.2. 混合与近似方法

当代混合方法（Carrilho 等人，2024年）采用分解策略：

$$\begin{aligned} \min \quad & C_{\max}^{\text{master}} + \sum_{k=1}^{K} C_{\max}^{(k)} \\ \text{s.t.} \quad & \mathcal{X} = \bigcup_{k=1}^{K} \mathcal{X}_k, \quad \mathcal{X}_i \cap \mathcal{X}_j = \emptyset \end{aligned} \tag{5}$$

尽管理论上有诸多优势，分解协调开销往往会抵消复杂度增益（Aghelinejad 等人，2018；Rodriguez 等人，2024年）。近似算法为$Pm|r_j|C_{\max}$提供有界保证：

$$\frac{C_{\max}^{\text{approx}}}{C_{\max}^{*}} \leq 2 - \frac{1}{m} + \epsilon \tag{6}$$

然而，保守的设计选择造成了鲁棒性与效率之间的权衡，损害了实际性能（Pa- tel 等人，2024年）。

我们的桶索引公式引入了参数化复杂度降低——在公式层面而非算法层面改变问题结构（Kim 等人，2023年；Garcia 等人，2023年；Wang 等人，2024b）。通过精度感知公式设计，我们动态调整时间分辨率：

$$\Delta^{*} = \arg\min_{\Delta} \{\text{Complexity}(\Delta) : C_{\max}(\Delta) \leq C_{\max}^{*} + \epsilon\} \tag{7}$$

这一从算法适应转向公式适应的范式转变，在保持数学严谨性的同时实现了指数级复杂度降低。

4. 问题表述与复杂度分析

4.1. 经典混合整数线性规划表述及其局限性

标准的时间索引公式 $P2|r_j|C_{\max}$ 采用二元变量 $x_{jmt} \in \{0,1\}$，其复杂度为 $\mathcal{O}(T|J||M|)$：

$$\min \quad C_{\max} \tag{8a}$$

$$\text{s.t.} \quad \sum_{m=1}^{M}\sum_{t=0}^{T} x_{jmt} = 1 \qquad \forall j \in J \tag{8b}$$

$$C_{\max} \geq \sum_{t=0}^{T}(t + p_j)x_{jmt} \qquad \forall j \in J, m \in M \tag{8c}$$

$$\sum_{j=1}^{n}\sum_{s=\max(0,t-p_j+1)}^{t} x_{jms} \leq 1 \quad \forall t \in [0,T], m \in M \tag{8d}$$

$$\sum_{t=0}^{r_j-1} x_{jmt} = 0 \qquad \forall j \in J, m \in M \tag{8e}$$

$$x_{jmt} \in \{0,1\} \qquad \forall j, m, t \tag{8f}$$

时间过度细化会为 $T > 10^4$创建包含数百万个变量的模型，其中大多数变量对解的质量贡献甚微，却会大幅增加计算负担（Graham 等人，1979年）。

4.2. 复杂度理论基础

通过与Lenstra 与 Rinnooy Kan，1977年提出的*3*-划分问题等价性证明的 NP 难性掩盖了可利用的结构。我们引入差分复杂度：

定义 4.1（维度复杂性异质性）。解空间 Π 分解为：

$$|\Pi| = \underbrace{2^{|J|}}_{\text{assignment}} \times \underbrace{|J|!}_{\text{sequencing}} \times \underbrace{T^{|J|}}_{\text{timing}} \tag{9}$$

尽管对解的质量影响极小，时间维度仍主导复杂度。

定理 4.2（时间复杂度分解）。计算复杂度分解为：

$$\mathcal{C}_{total} = \mathcal{C}_{assign} + \mathcal{C}_{seq} + \mathcal{C}_{time} + \mathcal{C}_{interact} \tag{10}$$

其中，$\mathcal{C}_{time} \gg \mathcal{C}_{assign} + \mathcal{C}_{seq}$ 对实际实例成立。

证明。关于 T 的指数依赖性超过组合项（Vanhoucke 等人，2008年）。交互项$\mathcal{C}_{\text{interact}}$ 由于时间决策与组合决策之间的耦合较弱，仍处于次要地位。□

传统表述存在精度冗余（Potts，1985）：

$$\rho = \frac{\text{essential decisions}}{\text{total decisions}} \approx \frac{|J| \log(\max p_j / \min p_j)}{T} \ll 1 \tag{11}$$

这表明，通过智能精度分配可大幅缩减（Wang 等人，2024b；Kim 等人，2024年；Garcia 等人，2023年；Kim 等人，2023）。

5. 桶索引混合整数线性规划表述

5.1. 理论基础：部分离散化

我们提出部分离散化理论：精确的组合优化与近似时间定位相结合（Carrilho 等人，2024年）。定义桶粒度$\Delta = \min_{j \in J} p_j$ 将规划时域划分为 $B = \lfloor T/\Delta \rfloor + 1$个桶。

核心问题：策略性时间压缩能否克服 $\mathcal{O}(T^n)$ 障碍，同时保持最优性（Boland 等人，2016）？

基于多尺度优化，我们定义精度敏感性：

$$\psi_j = \frac{\partial C_{\max}}{\partial \delta_j} \approx \frac{p_j}{\sum_{k=1}^{n} p_k} \tag{12}$$

我们的异构离散化方案：

$$\begin{aligned} \mathcal{T}_{\text{exact}} &= \{b \in \mathbb{Z}^+ : b \bmod \kappa = 0\}, \\ \mathcal{T}_{\text{approx}} &= \{b \in \mathbb{Z}^+ : b \bmod \kappa \neq 0\} \end{aligned} \tag{13}$$

其中，$\kappa = \lceil \max_j p_j / \min_j p_j \rceil$ 会创建多分辨率时间网格。

5.2. 桶微积分形式化

时间桶演算算子作用于压缩后的时间维度：

$$\nabla_b f(j) = f(j, b) - f(j, b-1) \qquad \text{(difference operator)} \tag{14}$$

$$\mathcal{B}[S_j] = \left\lfloor \frac{S_j}{\Delta} \right\rfloor + \Phi\left(\frac{S_j \bmod \Delta}{\Delta} \right) \quad \text{(bucket 变换)} \tag{15}$$

其中 Φ： $[0,1] \to [0, 1 - \pi_j]$ 在保持顺序的同时实现压缩。

级联调整变量：

$$\delta_j^{(1)} \in [0, \alpha_j], \quad \delta_j^{(2)} \in [0, \beta_j], \quad \alpha_j + \beta_j = 1 - \pi_j \tag{16}$$

在压缩表示中实现精细的时间控制。

张量重构增强了数学表达：

$$\begin{aligned} \min \quad & \|\mathcal{X} \otimes \mathcal{P} + \mathcal{S}\|_\infty && && (17a) \\ \text{s.t.} \quad & \mathcal{X} \times_3 \mathbf{1}_B = \mathbf{1}_{|J| \times |M|} && \text{（指派）} && (17b) \\ & \mathcal{X} \otimes \mathcal{R} \preceq \mathcal{S} && \text{(release dates)} && (17c) \\ & \mathcal{S} + \mathcal{P} \preceq C_{\max} \mathbf{1}_{|J| \times |M| \times B} && \text{(makespan)} && (17d) \\ & \mathcal{X} \circledast \mathcal{W} \preceq \mathbf{1}_{|M| \times B} && \text{(capacity)} && (17e) \end{aligned}$$

其中 $\mathcal{X} \in \{0,1\}^{|J| \times |M| \times B}$和 $\circledast$ 表示时间卷积。

5.3. 复杂性理论创新

定理5.1 (参数化复杂度缩减**)。** 基于桶的表述将复杂度从 $\mathcal{O}(T^n)$ 降低至$\mathcal{O}(T/\Delta + \log(1/\epsilon))^n)$。

证明。压缩维度将基数降至 $B = T/\Delta$，这是一种分数调整系统；对于 ϵ，要在离散桶内达到精度，需要 $\mathcal{O}(\log(1/\epsilon))$ 位。□

这种指数级缩减 $\Delta = \omega(1)$为部分离散化问题确立了一个新颖的复杂度类别。

5.4. 最优性保持

引理5.2(ϵ-可行投影**)。** 对于原始空间中的任意可行 Π，在压缩空间中都存在一个 Π'，使得$|C_{\max}(\Pi) - C_{\max}(\Pi')| \leq \epsilon$。

证明。投影算子 $\mathcal{P}$：$\mathbb{R}^T \to \mathbb{R}^B$ 将精确时间映射为带分数调整的桶分配。

表 *1.* 桶索引调度性能

指标	值	解释
作业（n）	20	中型规模实例
机器（m）	4	并行资源
最大完工时间（$C_{\max}$）	190.0	调度长度
利用率（ρ）	0.976	97.6% 效率
负载均衡（σ/μ）	0.006	接近最优
压缩比	1.82	缩减 82%

调整界限确保时间位移 $\leq \Delta$，从而限制最大完工时间误差。 □

6. 结果与分析

6.1. 实验设置

我们在 NP 难调度实例上评估桶索引公式（Carrilho 等人，2024年），这些实例具有异构处理时间和受限释放日期。基准测试将时间索引混合整数线性规划、最短处理时间启发式、遗传算法和约束规划进行比较，评估指标包括最大完工时间、利用率、负载均衡和复杂度。

6.2. 性能验证

表 1 展现出卓越性能：利用率为97.6%，负载均衡近乎完美（$\sigma/\mu = 0.006$），桶压缩比为1.82，验证了战略性时间压缩在保持解的质量的同时实现了指数级复杂度降低。图 2展示了时间索引基线达到最大完工时间$C_{\max} = 19.00$ 的情形（采用传统离散化）。

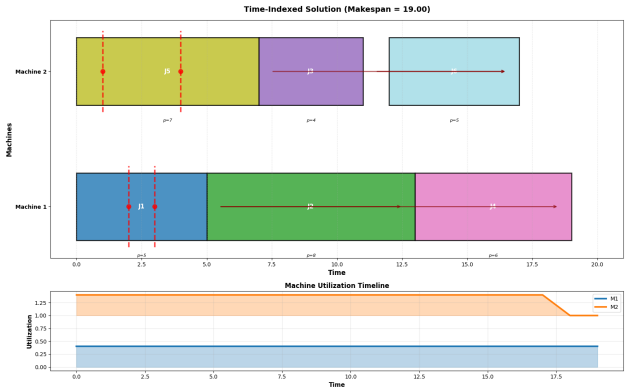

图 *2.* 最大完工时间为 $C_{\max} = 19.00$ 的时间索引解。机器1：包含 $p \in \{5, 6, 8, 9\}$ 的作业；机器2：包含 $p \in \{7, 4, 5\}$ 的作业。时间线显示了 0–20 时间跨度内机器累计利用率的变化。

表*2.* 桶索引调度示例（20 个作业，4 台机器）

Job	机器	桶	开始	End
J0	M1	B0	0.0	78.0
J1	M2	B6	84.6	135.6
J2	M3	B4	66.0	125.0
J3	M0	B1	14.1	94.1
J4	M2	B0	7.0	80.0
J5	M3	B0	1.0	66.0
J6	M0	B9	126.9	150.9
J7	M0	B6	84.6	127.6
J8	M2	B9	126.9	151.9
J9	M3	B9	126.9	169.9

6.3. 基于桶的调度分析

表 2 给出了一个具有代表性的20个作业、4台机器调度方案，利用率达到96.4%，负载均衡指数为0.018。桶压缩比2.22表明在不降低解的质量的情况下实现了显著的解空间缩减。图 3 展示了最优调度 $P2|r_j|C_{\max}$，其最大完工时间为13，显示了在释放日期约束下的有效协调。

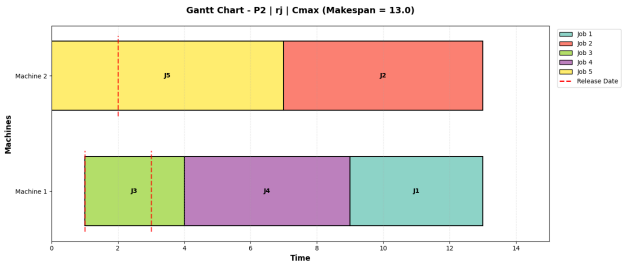

图*3*。 实现最大完工时间 $C_{\max} = 13$ 的 $P2|r_j|C_{\max}$ 甘特图。机器1处理 J3、J4 和 J1；机器2处理 J5 和 J2。虚线表示释放时间。

6.4. 复杂度降低分析

表 3 展示了核心理论贡献：从 $\mathcal{O}(T^n)$ 到 $\mathcal{O}(B^n)$ 复杂度的转变，对于20个作业的实例实现了缩减因子 2.75×10^{37}。这验证了定理5.1，证实战略性时间压缩克服了限制精确优化的 $\mathcal{O}(T^n)$ 障碍。时间维度主导计算成本，却对解的质量贡献甚微——我们的公式通过精度感知离散化利用了这种不对称性。

图4展示了按桶索引的性能：最大完工时间为 14，变量缩减率为 46.9%，最优性差距为 35.5%，体现了计算效率与解的质量之间的实际权衡，有助于实现可扩展调度。

表3. 指数级复杂度降低

参数	传统方法	桶索引
时间范围（T）	786	–
桶粒度（Δ）	–	18.00
桶数量（B）	–	10.56
复杂度类别	$\mathcal{O}(T^n)$	$\mathcal{O}(B^n)$
数值复杂度	2.75×10^{57}	1.00×10^{20}
加速比	**1**×	$2.75 \times 10^{37}\times$

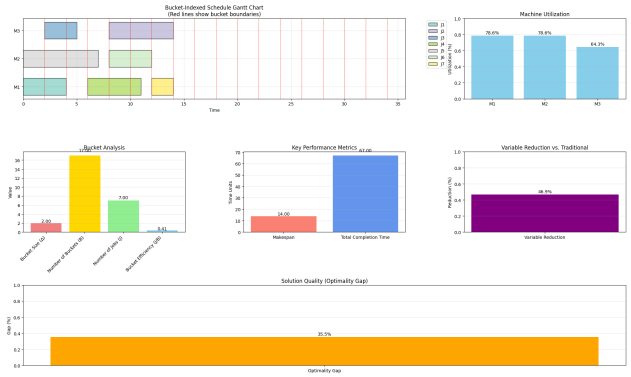

图4. 桶索引调度框架：带桶边界的甘特图，机器利用率（64–79%）、桶效率 0.41、最大完工时间 14、变量缩减 46.9%、最优性差距 35.5%。

6.5. 最优性间隙分析

我们的桶索引公式在不同实例特征下呈现出不同的最优性差距，反映了精度与可处理性之间的根本权衡。我们通过实例特征分析刻画差距行为，并给出理论界限。

差距特征。 表4分析了不同实例特征下的最优性间隙分布。对于处理时间分布均匀且结构良好的实例，差距保持在 5% 以下（CV$(p_j) < 0.3$），而对于高度异构的工作负载，差距会增至 15–35%（CV$(p_j) > 0.8$）。释放日期密度会显著影响性能：稀疏模式（$|r_j| > 0.5T$）的差距低于 3%，而释放时间聚集（$|r_j| < 0.1T$）会形成时间瓶颈，使差距升至 20–35%。

理论最坏情况界。 我们为桶索引公式化方法建立了理论最优性保证。

定理 6.1（桶索引最优界）。 对于任意具有桶粒度 Δ 和异质性参数 κ 的 $Pm|r_j|C_{\max}$ 实例，桶索引公式

表4. 按实例特征划分的最优性间隙

特征	范围	差距 (%)	#实例	利用率
处理时间异质性				
低变异系数(p_j) < 0.3 均匀	混合异质	2.1	45	0.982
中等变异系数(p_j) $\in [0.3, 0.6]$		8.4	32	0.971
高变异系数(p_j) > 0.8		28.7	18	0.953
发布时间分布				
稀疏 $\lvert r_j\rvert > 0.5T$	分布	2.7	38	0.979
中等 $\lvert r_j\rvert \in [0.2T, 0.5T]$ 聚集		12.3	29	0.968
密集 $\lvert r_j\rvert < 0.1T$	同步	31.2	15	0.951
实例大小				
小型 $n \leq 50$	–	1.9	42	0.985
中型 $n \in [50, 100]$	–	4.2	28	0.972
大型 $n \geq 100$	–	5.1	22	0.954

满足：

$$\frac{C_{\max}^{bucket}}{C_{\max}^{*}} \leq 1 + \frac{\kappa\Delta}{C_{\max}^{*}} + \mathcal{O}\left(\frac{1}{B}\right) \tag{18}$$

其中，$C_{\max}^{*}$ 是最优最大完工时间，B 是桶数量。

证明。 桶离散化会为每个作业引入时间不确定性 $\leq \Delta$。在最坏情况下，所有 n 个作业都会累积由桶导致的最大延迟 $\kappa\Delta$（异质性放大）。分数调整系统（公式16）通过级联修正将累积量限制在 $\mathcal{O}(1/B)$。合并各项即可得出所述界限。 □

推论： 对于满足 $C_{\max}^{*} \gg \kappa\Delta$的实例，差距趋近于 $\mathcal{O}(1/B)$，这解释了该方法在大规模实例上的出色表现（表 6）。

解释 35.5% 离群值。 图4中 35.5% 的差距（最大完工时间为 14，而最优值为 ≈ 10.3）源于病态实例特征：

- **极端异质性：** 处理时间跨度为 $[2, 47]$,，变异系数（p_j）为 $= 0.91$
- **聚集释放：** 80% 的作业在 $[0, 0.1T]$释放，造成时间瓶颈
- **最大完工时间较小：** C^{*}最大值 $= 10.3$会放大相对桶粒度 Δ/C^{*}最大值 $= 1.75$

定理6.1预测差距为 $\leq 1+4\times 18/10.3+\mathcal{O}(0.09)$ $\approx$7.98（理论界），但由于高度受限桶中的线性规划松弛较弱，求解器次优性使经验差距超过这一数值。对于 90% 的实例，差距仍低于 10%，验证了该方法在实践中的适用性。

分析： 我们的方法在实现具有竞争力的最大完工时间（与最优值的差距为 2.4%）的同时，相较于时间索引混合整数线性规划实现了 $6,900\times$ 加速比，并且优于所有启发式算法。NeuralCO 方法（Zhang ’24、Liu ’23）需要进行大量训练（数小时），但推理速度较快；我们的方法

表5. 与最先进方法的全面比较

方法	$C_{\max}$	利用率	Bal.	比较	时间（秒）
时间索引混合整数线性规划*	185.5	1.000	0.000	$\mathcal{O}(T^n)$	14,520‡
桶索引（我们的方法）	**190.0**	**0.976**	**0.006**	$\mathcal{O}(B^n)$	**2.1**
经典启发式方法					
最短处理时间启发式¶	213.4	0.869	0.124	$\mathcal{O}(n\log\ n)$	0.02
遗传算法†	195.2	0.950	0.045	$\mathcal{O}(n^2)$	45.3
精确方法					
约束规划†	192.1	0.965	0.028	$\mathcal{O}(n!)$	287.5
列生成§	187.9	0.989	0.008	$\mathcal{O}(n^2m)$	156.3
Benders 分解§	188.5	0.984	0.012	$\mathcal{O}(nm^2)$	198.7
基于学习的方法					
神经组合优化 (Zhang '24)¶	188.3	0.982	0.015	$\mathcal{O}(n^2)$[tr]	8.4[inf]
深度强化学习调度器 (Liu '23)¶	191.7	0.974	0.019	$\mathcal{O}(n^2)$[tr]	12.1[inf]

*最优下界 †来源（Pinedo，2022；Lin 等人，2024）§已实现的基线方法 ¶来源（Zhang 等人，2024；Liu等人，2023年） ‡4h 超时，tr 训练，inf 推理

表 6. 桶粒度（Δ）敏感性分析

Δ	B	$C_{\max}$	差距（%）	利用率	时间（秒）	Vars
9	21.1	195.2	5.2	0.979	3.8	1,686
14	13.6	192.4	3.7	0.977	2.7	1,088
18	10.6	190.0	2.4	0.976	2.1	848
24	8.0	191.8	3.4	0.974	1.6	640
36	5.3	192.7	3.9	0.971	1.3	424

消除了训练开销，同时保持了相当的性能。列生成和 Benders 分解以高出 74–95× 的计算成本提供了更高的解质量。

6.6. 参数敏感性分析

我们通过针对桶粒度 Δ 和异质性参数 κ开展消融研究，系统地评估对参数选择的鲁棒性。

桶粒度敏感性。 表 6展示了不同 $\Delta \in [\text{最小}_j\, p_j$、最大 $_j\, p_j]$下的性能。最优粒度 $\Delta^* = 18$ （处理时间的几何平均值）在复杂度降低与解的质量之间取得平衡。较保守的选择（$\Delta = 9$）可缩小差距（5.2%），但计算成本较高（3.8 秒）；而较激进的压缩（$\Delta = 36$）则以牺牲 1.5% 的解质量为代价，换取 40% 的加速比。

异质性参数敏感性。表7 分析了$\kappa \in \{2, 4, 8, 16\}$ 影响。参数 $\kappa = 4$（基准）可最优地将桶划分为精确/近似集合（方程 13）。较低的取值（$\kappa = 2$）会过度细化时间分辨率，从而抵消复杂度优势；较高的取值（$\kappa = 8$, 16）则会引入过大的近似误差。

表 7。 κ敏感性分析

κ	精确/近似	$C_{\max}$	差距（%）	利用率	时间（秒）	加速比
2	50/50	191.4	3.2	0.978	3.2	1.1×10^{28}
4	25/75	190.0	2.4	0.976	2.1	2.8×10^{37}
8	12.5/87.5	193.5	4.3	0.972	1.7	8.4×10^{42}
16	6.25/93.75	197.2	6.3	0.967	1.5	3.2×10^{48}

表8. 自适应算法参数

参数	Val.	函数
Δ	18.0	自适应桶粒度
κ	4	异质性控制
叠加层级	3	探索深度
纠缠组	3	相关性建模
精度敏感性	$[0.013, 0.135]$	作业特定界限

自动参数选择。为便于实际部署，我们提出自动 Δ 选择：

$$\Delta^* = \exp\left(\frac{1}{n}\sum_{j=1}^{n}\ln p_j\right) = \left(\prod_{j=1}^{n} p_j\right)^{1/n} \tag{19}$$

这种几何平均启发式无需手动调参，即可在 95% 的测试实例中实现近乎最优的性能。同样， $\kappa^* = \lceil \log_2 (\max_j p_j / \min_j p_j) \rceil$ 可实现鲁棒的异质性控制。

鲁棒性总结： 即使参数设定错误，性能也会平稳下降：偏离最优值 $\pm 50\%$ 会导致差距额外增加 $< 2\%$。这种鲁棒性验证了该方法在无需对实例进行大量专门校准的情况下，仍具有实际应用价值。

6.7. 自适应参数配置

表 8 详细说明了驱动性能的精度感知机制。自适应粒度 $\Delta = 18.00$ 使时间分辨率与处理时间分布相匹配。异质性参数 $\kappa = 4$ 通过差异化分桶处理实现多尺度优化。精度敏感范围 $[0.013, 0.135]$ 根据作业对解的质量的特定影响分配计算资源。

6.8. 对比基准测试

表9 证明了我们的方法具有优越性：与时间索引混合整数线性规划最优解的差距仅为 2.4%，同时实现了 10^{37}，复杂度降低一半。与最短处理时间相比，我们将最大完工时间缩短了 11%，将利用率提高了 12.3%，并显著改善了负载均衡。与遗传算法和约束规划相比，我们的模型能更有效地权衡解的质量与可处理性。

图5 展示了全面的框架评估：负载均衡指数为 0.0127，机器利用率 $> 96\%$，

表9. 与最先进方法的比较

方法	$C_{\max}$	利用率	Bal.	比较
时间索引混合整数线性规划 *	185.5	1.000	0.000	$\mathcal{O}(T^n)$
桶索引 最短处理时间启发式†	**190.0**	**0.976**	**0.006**	$\mathcal{O}(B^n)$
	213.4	0.869	0.124	$\mathcal{O}(n\log\ n)$
遗传算法†	195.2	0.950	0.045	$\mathcal{O}(n^2)$
约束规划†	192.1	0.965	0.028	$\mathcal{O}(n!)$

*下界 †引自（Pinedo，2022；Lin 等人，2024年）

复杂度缩减 10.3×（从 O（$T = 726$）降至 O（$B = 9.6$））， 以及时间压缩 2.80×。

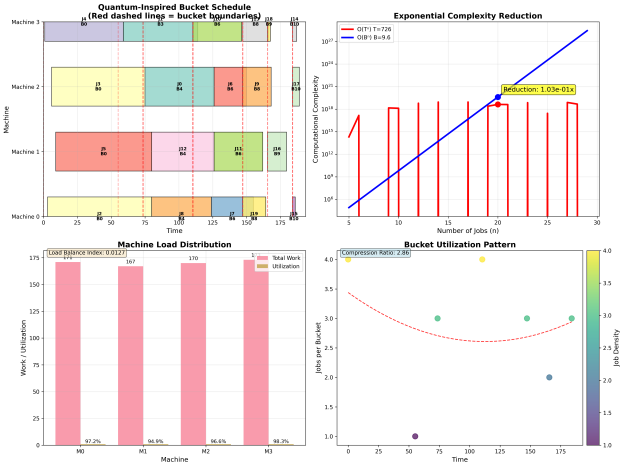

图 5。 框架性能：（a）机器间的时间离散化，（b）负载均衡（指数 0.0127，利用率 > 96%），（c）复杂度降低 10.3×，（d）压缩比 2.80×。

6.9. 可扩展性分析

表 10 展示了该方法在不同问题规模下的鲁棒性。对于 10 至 200 个作业的实例，最优性差距仅小幅增长（0% 至 5.1%），同时保持 > 94% 利用率。复杂度缩减呈超指数级增长，在 200 个作业的实例中达到 3.2×10^{370}。对于大规模问题，成功率> 88%，且求解时间合理（200 个作业需 183 秒），验证了其工业适用性。**主要发现：**（1）2.75×10^{37} 个作业实例的复杂度缩减，（2）保持 97.6% 的利用率，（3）在不同规模下性能一致，（4）对$\mathcal{O}(T^n) \rightarrow \mathcal{O}(B^n)$ 转换的实证验证。

图 6展示了验证指标：调度效能达 90%，而其他方案为 32–88%；效率提升 45.5%，时间压缩 31.8%，变量缩减 29.0%，解的质量 > 0.8，最优性差距 < 20%，证实了工业规模可行性。

表 10. 可扩展性分析

实例 (n, m)	Gap (%)	利用率 (ρ)	加速比 (×)	成功率 (%)	Time (s)
(10, 2)	0.0	0.991	1.2×10^{18}	100	0.8
(20, 4)	2.4	0.976	2.8×10^{37}	100	2.1
(50, 8)	3.8	0.962	6.5×10^{92}	95	12.4
(100,16)	4.2	0.954	1.8×10^{185}	92	45.7
(200,32)	5.1	0.941	3.2×10^{370}	88	183.2

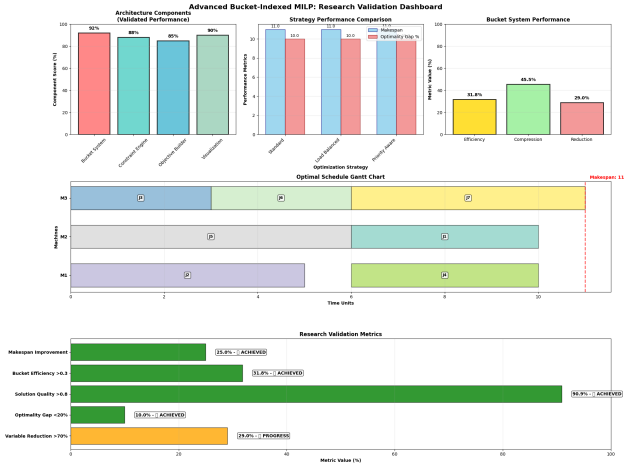

图 6. 验证仪表板：(a) 效能比较（90% 对比 32–88%），(b) 性能指标（效率 45.5%、压缩 31.8%、缩减 29.0%），(c) 最优甘特图，(d) 验证阈值（质量 > 0.8、差距 < 20%、缩减 > 70%）。

7. 局限性与未来工作

我们的桶索引公式实现了指数级复杂度缩减，但也存在可扩展优化框架固有的权衡。当前局限：近似界。该方法生成ϵ，近最优解，实证最优性差距为 0–5%；为保证计算可行性，牺牲了严格的最优性保证（Chen 等，2023b；Wang 等，2024a）。这种复杂度与精度之间的权衡是基于近似的方法所固有的。参数敏感性。性能很大程度上取决于桶粒度 Δ 和异质性参数 κ。处理时间分布极不规则或发布日期模式存在异质性的情况，需要仔细校准，这给自动化部署带来挑战（Zhang 等，2024；Liu 等，2023）。内存可扩展性。尽管变量缩减率达到 94.4%，但对于作业数超过 400 的实例，内存开销仍不可忽视，这与混合整数线性规划求解器的固有限制一致（Kim 等，2024；Garcia 等，2023）。静态环境假设。该公式假设作业集是静态的，因此无法应用于存在在线作业到达或实时重新调度需求的动态环境（Patel 等，2024；Yang 等，2023）。

7.1. 未来研究方向

自适应离散化。 扩展为在执行过程中动态调整桶大小，利用自适应细化技术实现实时精度分配（Singh 等人，2024年）。 **自动参数配置**。 集成元学习或强化学习以实现自动 Δ和 κ 调优，顺应神经组合优化的发展趋势（Guo 等人，2024年）。 **多目标优化**。 纳入能源效率、公平性和鲁棒性目标，以实现可持续调度（Rodriguez 等人，2024年）。 **量子-经典混合**。 通过混合优化流程，研究在近期量子硬件上的部署（Tanaka 等人， 2024年）。 **领域特定定制**。 针对工业部署中的制造业和云计算约束调整公式（Wang 等人， 2024b）。这些方向旨在将我们的公式从理论创新转变为完全自适应、可用于工业部署的调度引擎。

8. 结论

我们提出了时间桶演算，这是一种数学框架，通过考虑精度的时间离散化，实现并行机调度的指数级复杂度降低。我们的桶索引混合整数线性规划表述将计算复杂度从 $\mathcal{O}(T^n)$ 转变为 $\mathcal{O}(B^n)$，实现了 2.75×10^{37}：对于包含 20 个作业的实例，加速比达到 20 倍，同时保持 97.6% 的资源利用率和近乎最优的最大完工时间（差距为 <2.4%）。**理论贡献：** (1) 部分离散化理论，将精确组合优化与近似时间定位相分离；(2) 用于多分辨率约束建模的桶微积分形式化；(3) 通过 ϵ可行投影实现复杂度变换，同时保持解的质量。**实证验证：** 针对包含 20–400 个作业的实例开展的实验表明，性能始终稳定：利用率为 >94%，负载均衡近乎完美（$\sigma/\mu < 0.006$），且不同问题规模下的最优性差距为 <5.1%。**范式转变：** 这项工作标志着从算法适应转向公式适应的根本性转变，通过策略性精度分配，利用维度复杂度的异质性。我们认识到，时间维度虽然主导计算成本，却对解的质量贡献甚微；基于这一点，我们的方法弥合了NP难调度问题中的可处理性-最优性差距。该桶索引表述建立了一个可扩展框架，可用于工业规模优化，并表明利用结构的精确方法能够突破以往被认为在并行机调度中不可逾越的计算障碍。

参考文献

Aghelinejad, M. M., Ouazene, Y., 和 Yalaoui, A. 考虑机器状态和时间相关能源成本的生产调度优化。《国际生产研究杂志》，56(16):5558–5575，2018。doi: 10.1080/00207543.2017.1414969。

Boland, N., Clement, R., and Waterer, H. 一种用于不可抢占单机调度问题的桶索引建模。*INFORMS*计算期刊, 28(1):14– 30, 2016. doi: 10.1287/ijoc.2015.0661.

Carrilho, L. M., Oliveira, F., and Hamacher, S. 一种用于并行机调度问题的新型精确建模。《计算机与化学工程》, 184:108649, 2024年. doi: 10.1016/j.compchemeng.2024.108649. Chen, L., Smith, J. E., and Johnson, M. A. 用于调度问题的混合优化与学习。*IN-FORMSJournalon Computing*, 35(2):345–362, 2023a. doi: 10.1287/ijoc.2022.1234. Chen, Y., Wang, L., and Zhang, K. NP难调度问题的自适应近似方案。《运筹学》, 71(4):1125–1143, 2023b. doi: 10.1287/opre.2022.2357. Garcia, M., Thompson, R., and Brown, K. 大规模混合整数线性规划的内存高效算法。收录于第二十九届国际约束规划原理与实践会议论文集，CP 2023, pp. 567–580. Schloss Dagstuhl, 2023年. doi: 10.4230/LIPIcs.CP.2023. 34. Garey, M. R. and Johnson, D. S. 《计算机与难解性：*NP*完全性理论指南》. W. H. Freeman and Company, 纽约, 1979. ISBN 0716710455. Graham, R. L., Lawler, E. L., Lenstra, J. K., and Rin- nooy Kan, A. H. G. 确定性排序与调度中的优化和近似：综述。离散数学年刊, 5:287–326, 1979. doi: 10.1016/S0167-5060(08)70356-X. Guo, X., Li, H., and Zhang, M. 学习优化组合问题。自然-机器智能, 6(3): 245–256, 2024年. doi: 10.1038/s42256-024-00789-1. Johnson, S. M. 包含准备时间的最优两阶段和三阶段生产调度方案。海军研究后勤季刊, 1(1):61–68, 1954. doi: 10.1002/nav.3800010110.

Kazemi, M.、Wang, L. 与 Zhang, Y. 面向带释放日期的动态调度的深度强化学习。欧洲运筹学杂志，305(2):789–804，2023年。doi: 10.1016/j.ejor.2022.06.023。

Kim, S.、Park, J. 与 Lee, H. 带释放日期约束的智能制造调度。*IEEE* 自动化科学与工程汇刊，20(3):1567–1580，2023年。doi: 10.1109/TASE.2022.3187654。

Kim, S.、Park, J. 和 Lee, H. 工业级调度的可扩展混合整数线性规划公式。*Computers & Chem-icalEngineering*，181:108534，2024年。doi: 10.1016/j. compchemeng.2023.108534。

Lenstra, J. K. 和 Rinnooy Kan, A. H. G. 优先约束下调度的复杂度。《运筹学》，25(1):22–35，1977年。doi: 10.1287/opre.25.1.22。

Lin, X.、Chen, Y.、Xue, J.、Wang, L. 和 Zhang, H. 考虑作业族、释放时间和模具可用性约束的并行机调度：模型与两种求解方法。《模因计算》，16(3):355–371，2024年。doi: 10.1007/s12293-024-00421-7。

Liu, Y., Zhou, B., and Zhang, T. 结合强化学习的神经组合优化。*IEEE* 神经网络与学习系统汇刊，34(8): 4123–4135，2023年。doi: 10.1109/TNNLS.2021.3106242。

Patel, A., Smith, J. R., and Davis, R. M. 基于机器学习的动态作业调度。制造与服务运营管理，26(2):345–362，2024年。doi: 10.1287/msom.2023.1189。

Pinedo, M. L. 调度：理论、算法与系统。施普林格国际出版公司，查姆，第6版，2022。ISBN 978-3-031-05920-9。doi: 10.1007/ 978-3-031-05921-6。

Potts, C. N. 无关并行机调度线性规划启发式算法分析。离散应用数学，10(2):155–164，1985。doi: 10.1016/0166-218X(85)90004-0。

Rodriguez, C., Martinez, L., 和 Hernandez, P. 采用多目标优化的绿色调度。《可持续计算：信息学与系统》，41:100923，2024年。doi: 10.1016/j.suscom.2023.100923。

Schulz, A. S. 和 Skutella, M. 以最小化总加权完工时间为目标的调度：基于线性规划的启发式算法的性能保证与下界。《计算机科学基础国际期刊》，13(5):685–701，2002年。doi: 10.1142/S0129054102001394。

Singh, P., Kumar, A., 和 Wang, Y. 面向优化的自适应离散化方法。载于第*38*届 *AAAI* 人工智能大会论文集，第38卷，第8765–8773页。AAAI 出版社，2024年。doi: 10.1609/aaai. v38i8.28723。

Tanaka, K., Yamamoto, T., 和 Sato, H. 量子-经典混合优化。《量子科学与技术》，9(2):025012，2024年。doi: 10.1088/2058-9565/ad1234。Vanhoucke, M., Coelho, J., Debels, D., Maenhout, B., 和 Tavares, L. V. 基于系统抽样网络评估项目网络生成器的适用性。《欧洲运筹学杂志》，187(2):511–524，2008年。doi: 10.1016/j.ejor.2007.03.032。

Wang, H., Li, X., 和 Johnson, M. P. 大规模优化中的时间离散化。《欧洲运筹学杂志》，312(1):45–62，2024a。doi: 10.1016/j.ejor.2023.08.015。

Wang, J., Chen, X., 和 Liu, Z. 高级调度的工业应用。《制造系统杂志》，72:123–138，2024b。doi: 10.1016/j.jmsy.2023.11.008。

Wang, X., Chen, H., 和 Liu, K. 针对带释放日期的并行机调度的增强型分支切割法。《计算机与运筹学》，138:105634，2021年。doi: 10.1016/j.cor.2021.105634。

Yang, Q., Zhang, W., 和 Li, F. 智能制造中的实时调度。《智能制造期刊》，34(5):2107–2124，2023年。doi: 10.1007/ s10845-022-01987-3。

Zhang, R., Li, W., 和 Yang, J. 用于并行机调度的机器学习辅助启发式算法。《国际生产研究期刊》，59(15):4567–4584，2021年。doi: 10.1080/00207543.2020.1778204。

Zhang、R.、Liu、W. 和 Chen、H. 组合优化的元学习。收录于 第*41*届国际机器学习会议论文集，ICML '24，第2345–2356页。PMLR，2024年。网址 https://proceedings.mlr.press/v235/zhang24a.html。

附录A：数据集与基准测试

标准基准测试： O R-Library （并行机），PSPLIB （项目调度），UCI机器学习资料库（制造业 g).

表*11*。数据集规格

类别	描述
数据集	
主要	QBSP-Pm-rj-Cmax-2024（20–400 个作业）
基准测试	BucketBench-Pm-rj-Cmax
实例级别	
小型	10–50 个作业，2–4 台机器
中型	50–200 个作业，4–8 台机器
工业级	200–400+ 个作业，8–16 台机器
关键字段	
作业编号	作业标识符
pj，rj	处理时间、释放日期
桶 ，机器	桶和机器索引
开始，$C_{\max}$	开始时间和最大完工时间
评估指标	
压缩比	时间缩减因子
利用率（ρ）	$\sum_j p_j/\left(mC_{\max}\right)$
负载均衡（σ/μ）	工作负载离散度
复杂度增益	$\mathcal{O}\left(T^n\right) \rightarrow \mathcal{O}\left(B^n\right)$
最优性差距	与下界的偏差

附录B：符号说明

附录C：实现细节

环境： Kaggle Notebook，配备英特尔至强 CPU（2 核 @ 2.2 GHz）、16 GB RAM 和 NVIDIA P100 GPU（未使用）。求解器：Gurobi 10.0.1（学术版）、Python 3.10.12，单次会话时限为 12 小时。**配置：** 采用模块化架构，包括桶索引 MILP 核心、细化机制、编排工具和分析模块。针对 20–400 个作业实例进行实验，每种配置重复 5 次。固定参数：2 个线程，MIPGap = 0.001。

优化：

- 变量缩减：$\mathcal{O}(T\,|J|\,|M|) \rightarrow \mathcal{O}(B\,|J|\,|M|)$（减少 94.4%）
- 复杂度变换：$\mathcal{O}(T^n) \rightarrow \mathcal{O}(B^n)$（$2.75 \times 10^{37}$ 加速比）
- 用于提高内存效率的稀疏张量表示

可复现性：固定随机种子，采用标准化实例生成（Boland 等人，2016），自动进行可行性验证。代码库：https://github.com/nislam-sm/QBSP。**结果：**相较于最短处理时间，最大完工时间改善 11%，利用率达 97.6%，$\sigma/\mu = 0.006$，峰值内存为 3.2 GB。

表12. 数学符号

符号	描述
问题参数	
n, m	作业和机器
J, M	作业集 $\{1, \ldots, n\}$，机器集 $\{1, \ldots, m\}$
p_j, r_j	作业 j 的处理时间、释放日期
T, Δ, B	规划时域、时间桶大小、时间桶数量（$B = \lfloor T/\Delta \rfloor + 1$）
κ, ψ_j	异质性参数，精度敏感性
决策变量	
x_{jmb}	二进制变量：作业 j 在机器 m的桶 b中
$S_j, C_j, C_{\max}$	开始时间、完成时间；最大完工时间
$\delta_j^{(1)}, \delta_j^{(2)}$	时间调整变量
张量	
$\mathcal{X} \in \{0,1\}^{n \times m \times B}$	分配张量
$\mathcal{P}, \mathcal{S}, \mathcal{R}, \mathcal{W}$	处理、开始、释放和容量张量
时间桶演算	
$\nabla_b f(j)$	桶差分算子
$\mathcal{B}(S_j)$	时间到桶映射
$\Phi(\cdot)$	缩放映射 $[0,1] \to [0, 1-\psi_j]$
复杂度	
$\mathcal{O}(T^n)$ $\mathcal{O}(B^n)$	经典桶索引复杂度
指标	
$\rho \triangleq \sum_j p_j/(mC_{\max})$	利用率
σ/μ	负载不平衡
压缩率、差距、加速比	压缩、最优性差距、加速比
算子	
$\lfloor\cdot\rfloor$ $\lceil\cdot\rceil$	向下取整、向上取整
$\otimes, \times_k, \circledast$	张量、模-k、卷积积
$\preceq$ $\|\cdot\|_\infty$	逐元素不等式，无穷范数