arXiv:2611.29883v1 [cs.DS] 23 Nov 2026

无关机器调度的可信建议

Baruch Schieber HadasShachnai Tami Tamir 新泽西理工学院 以色列理工学院 赖希曼大学 新泽西州纽瓦克 美国 海法，以色列 赫兹利亚，以色列sbar@njit.edu hadas@cs.technion.ac.il tami@runi.ac.il

摘要

数十年来，在标准目标下对无关并行机进行调度，例如最小化最大完工时间或实现最大最小公平性，一直是理论上的重大挑战。一个突出的例子是 Lenstra、Shmoys 和 Tardos（1990）针对 Min Makespan 问题确立的长期存在的 2 倍近似障碍。我们在可信建议算法的框架下研究这些基础性挑战。我们考虑这样一种设定：访问可信预言机一次，由其揭示有关最优解的部分但完全准确的结构信息。该框架使我们能够量化各种可信建议带来的近似保证提升。

我们形式化定义了一系列可信建议预言机，从能够揭示长作业精确分配方案的强预言机，到仅指明哪些机器处理此类作业的弱预言机。利用这些信息，我们获得了突破经典最坏情况界限的近似保证。具体而言，我们证明，即使是最弱的预言机，也能将 MinMakespan 的近似比提升至$\sqrt{3} \approx 1.732$。我们还分别为Restricted Assignment和 Graph Balancing取得了改进后的近似比$\frac{\sqrt{33}+1}{4} \approx 1.686$ 和 $\frac{5}{8}\sqrt{7} \approx 1.654$ 。对于最大最小公平性目标（即 Santa Claus 问题），使用弱预言机可获得 2-近似。最后，我们证明，即使采用所考虑的可信建议预言机，这些问题仍然是 NP困难的，无法在某些常数因子范围内实现近似。有趣的是，对于 Santa Claus问题，我们证明其下界为 2，且该界是紧的，与经典的无预言机界相匹配。

1 引言

在无关并行机上针对最大完工时间最小化或最大最小公平性等标准目标进行调度，是组合优化中的一大挑战。数十年来，这些问题的经典近似算法始终面临难以突破的理论障碍。一个典型例子是 Min Makespan 问题，在标准调度记法中记作 $R||C_{\max}$。输入包括一组 n 作业、一组 m 机器，以及一个 $n \times m$矩阵 p，其中 $p_{ij} > 0$ 表示作业 j 在机器 i上的处理时间。目标是将所有作业调度到机器上，使任一机器的最晚完工时间（即最大完工时间）最小。Lenstra、Shmoys 和 Tardos 的经典线性规划（LP）舍入技术 [15] 可实现 2-近似，这一界限至今仍 notoriously 难以突破。对此问题研究较为充分的类别包括 Restricted Assignment 问题（其中 $p_{ij} \in \{p_j, \infty\}$）及其特例 Graph Balancing 问题，在该问题中，每个作业最多可分配到两台机器。另一种研究颇多的变体是最大最小公平分配（即 Santa Claus 问题），其目标是最大化任一机器上的最小负载。

本文在可信建议算法框架下探讨这些经典调度问题。我们证明，关于最优解的有限但准确的信息，能够带来突破传统最坏情况障碍的近似保证。具体而言，除问题输入外，算法还会从一个可访问固定最优解的预言机处获得有限的正确信息。由于预言机只揭示部分信息，算法仍必须计算出完整的调度。我们所有的算法都只访问一次预言机。对预言机的有限访问以及它所揭示的部分信息，都要求我们的算法确保所获取的信息确实有用。

我们对预言机层级进行了形式化定义，涵盖从能够揭示部分作业精确分配的强预言机，到仅提供最少聚合信息的弱预言机。我们分析了这些部分信息对近似保证的影响。

关于可以向预言机查询哪些信息，决策至关重要。已知的困难性结果表明，例如，揭示 Min Makespan问题中作业的实际处理时间并不能带来显著优势，因为这些信息在 Restricted Assignment 问题中本来就是已知的。通过引入极小的虚拟作业，很容易看出，了解分配给每台机器的作业确切数量同样无济于事。此外，知道最优最大完工时间的值 C^*_{max} 也毫无用处，因为目前已知的所有下界仍然成立。这些下界均基于从三维匹配或 3-SAT 出发的归约；在这类归约中，知道存在完美匹配或满足赋值并不会使问题变得更简单。我们选择这样定义预言机，使其能够改进目前已知的最佳近似保证；与此同时，正如我们在第 6 节所示，不可近似界仍然成立。

研究的问题。 无关并行机上的 Min Makespan 实例由一组作业$J = \{1, \ldots, n\}$ 和一组机器 $M = \{1, \ldots, m\}$组成，并给定每个作业j 在机器 i、 $1 \le j \le n$、 $1 \le i \le m$上的处理时间 $p_{ij} \in \mathbb{Z}^+$。目标是将作业分配给机器，使最大完工时间最小；最大完工时间是指分配给某台机器的总处理时间的最大值。形式化地说，分配是一个函数 $\sigma\colon J \to M$，最优最大完工时间为 $C^*_{max} = \min_\sigma \max^{i \in M} \sum_{\{j \in J:\ \sigma(j) =i\}} p_{ij}$。

restricted assignment 问题是 Min Makespan 的一个特殊情形，其中 $p_{ij} \in \{p_j, \infty\}$，对于

所有 $1 \le j \le n$ 和 $1 \le i \le m$均成立。 RESTRICTED ASSIGNMENT 的一个特殊情形，即每个作业最多可分配到两台机器，称为 GRAPH BALANCING。该问题可以表述为如下图问题。考虑无向边加权图 $G = (V, E, p)$，其中 V表示机器集合。每条边 $e = u, v \in E$ 对应一个只能在机器 u和 v上处理的作业；该作业在任一台机器上的处理时间为 p_e，在其他所有机器上的处理时间为 ∞。自环对应只能分配给单台机器的作业。目标是确定边的一个定向，使所有顶点中的最大负载 最小，其中顶点的负载定义为其入边权重之和。显然，最大负载对应于最大完工时间。

SANTA CLAUS 问题的输入与 MIN MAKESPAN 的输入相同：一组 M m机器、一组 J n作业，以及一个处理时间矩阵 $n \times m$。目标是以“公平”的方式将作业分配给机器，使任意机器上的最小负载（总处理时间）最大化。形式上，分配是一个函数 σ：$J \to M$，最优值为 $C^*_{min} = \max_\sigma \min^{i \in M} \sum_{\{j \in J:\ \sigma(j) = i\}} p_{ij}$。这个最大最小公平分配 问题记作 SANTA CLAUS，因为作业可以对应礼物，机器可以对应孩子，处理时间可以对应估值，而目标则是以一种能最大化最不满意孩子估值的方式将礼物分给孩子。

1.1 我们的贡献

本文的一项主要贡献是引入了不同的预言机，它们能够揭示关于某个特定最优解 OPT 的有用信息。如下文所述，即使在我们最弱的预言机下，所有研究的问题都能获得更优的近似比；然而，在这一最弱预言机下，这些问题仍然是 NP困难的，无法近似求解。

对于 $r \ge 1$，我们称 $\mathcal{A}$ 是一个用于求解最小化（最大化）问题Π 的 r-近似算法，当且仅当对于 Π的任意实例 I， $\mathcal{A}(I)/val_{\mathrm{O}}(I) \le r$ $(val_{\mathrm{O}}(I)/\mathcal{A}(I) \le r)$，其中 $val_{\mathrm{O}}(I)$是 I的最优解的值。我们首先定义不同的预言机，然后分别介绍针对每种预言机的结果。

给定阈值 $\ell > 0$，如果一个作业在 OPT 中的处理时间超过 ℓ，我们就称其为长作业。我们首先考虑长作业调度预言机 $\mathsf{Oracle_{LS}}$，它会揭示 OPT 中所有长作业的分配。作为一个简单的热身，我们证明，对于任意 $\epsilon \in (0, 1],$，访问 $\mathsf{Oracle_{LS}}$ 都能为所考虑的所有问题提供$(1+\epsilon)$-近似解。

我们的下一个预言机是长作业代表预言机，记为 $\mathsf{Oracle_{LR}}$，它会揭示长作业的一个代表性子集的分配。具体而言，对于每台在 OPT 中至少处理一个长作业的机器 $i \in M$，该预言机会揭示其中恰好一个此类长作业的身份。我们证明，$\mathsf{Oracle_{LR}}$ 对于 MIN MAKESPAN的近似比为 $\frac{1+\sqrt{5}}{2} \approx 1.618$，对于 RESTRICTEDASSIGNMENT的近似比为 $\frac{1}{2} + \frac{3}{4}\sqrt{2} \approx 1.561$。

我们最弱的预言机，即长作业机器预言机 $\mathsf{Oracle_{LM}}$，会揭示在 OPT 中处理长作业的机器子集 M_L。这个弱预言机不会指出分配给这些机器的任何作业。我们证明，$\mathsf{Oracle_{LM}}$ 仍能使我们将 MIN MAKESPAN 的经典界限 2 改进至 $\sqrt{3} \approx 1.732$。对于 RESTRICTED ASSIGNMENT 和 GRAPH BALANCING，分别可得到更好的界限 $\frac{\sqrt{33}+1}{4} \approx 1.686$ 和 $\frac{5}{8}\sqrt{7} \approx 1.6536$。对于 SANTA CLAUS 问题，我们证明其比率为 2。

表1总结了我们的结果。对于 $\mathsf{Oracle_{LS}}$，较小的阈值 ℓ 会带来更多作业分配。因此，我们能够任意接近 OPT，这并不令人意外。相比之下，对于较弱的预言机，选择参数 ℓ 是一项关键挑战。如果 ℓ 过低，我们可能无法利用已揭示的

	预言机$_{\mathsf{LS}}$	预言机$_{\mathsf{LR}}$	预言机$_{\mathsf{LM}}$
Min Makespan(2)	$\mathbf{1+\epsilon}$	**1.618**	**1.732**
Restricted Assignment(2)	$\star$	$\star$	**1.686**
Graph Balancing(1.75)	$\star$	**1.561**	**1.654**
Santa Claus($O(n^c)$)	$\mathbf{1+\epsilon}$	$\star$	**2**

表1：本文给出的界以粗体显示。每行括号中注明了多项式时间算法可实现的已知最佳无预言机界。标有 $\star$ 的条目可由更一般的问题或更弱的预言机推导得出。

信息。该表展示了预言机通过一次预言机访问并采用精心选择的阈值所能达到的近似比。

即使使用最强预言机 Oracle$_{\mathsf{LS}}$，在阈值$\ell = \Omega(C^*_{max})$的假设下，我们研究的所有问题仍然是 NP困难的。这是因为，即使在没有作业长度超过该阈值的实例中，所有问题也都是 NP困难的。我们证明，使用最弱的预言机 Oracle$_{\mathsf{LM}}$时，Min Makespan 和 Santa Claus 的图平衡变体也难以近似。回顾一下，在图平衡变体中，每个作业最多可分配给两台处理时间相同的机器。我们假设预言机会揭示最优解的值（Min Makespan 的值为 C^*_{max}，SantaClaus的值为 C^*_{min}），以及被分配了长度超过该值的 α 倍的作业的机器集合，其中某个$\alpha \in (0,1)$。我们证明，对于 $\alpha \in (0, \frac{1}{2}]$，Min Makespan的图平衡变体在近似比优于 $\frac{3}{2}$时是 NP困难的；对于 $\alpha \in (\frac{1}{2}, 1)$，近似比优于 $2-\alpha$时也是如此。我们还证明，对于 $\alpha \in (0, \frac{1}{2}]$，Santa Claus 的图平衡变体在近似比优于 2 时是 NP困难的；对于 $\alpha \in (\frac{1}{2}, 1)$，近似比优于 $\frac{1}{\alpha}$ 时也是如此。我们的 2-近似 SantaClaus 算法会查询 Oracle$_{\mathsf{LM}}$，以获取长度至少为 $\frac{1}{2}$的作业相关信息，因此其性能达到了下界。

组织结构。在本节余下部分中，我们将重点介绍所用技术，并概述相关工作。第 2 节介绍我们关于 Min Makespan 的结果，第 3 节给出针对 Restricted Assignment的改进界。第 4 节和第 5 节分别讨论 Graph Balancing 和 Santa Claus。我们的困难性结果见第 6 节。最后，第 7 节总结全文并展望未来研究方向。

1.2 技术

借助预言机的最大完工时间最小化。我们的主要技术贡献是利用长作业机器预言机 Oracle$_{\mathsf{LM}}$ 推导出更优的近似保证。我们将这一技术应用于 Min Makespan 和 Restricted Assignment。下文概述针对 Min Makespan 的方法（见第 2 节）。

为确定预言机查询中哪些作业被视为长作业，我们如下选择阈值 $\ell > 0$。利用 [15] 的线性规划解，我们推导出最优最大完工时间 C^*_{max} 的下界 T。接着，我们引入参数 $\alpha \in (0,1]$，，其取值将在后文优化），并以阈值 $\ell = \alpha T$ 查询 Oracle$_{\mathsf{LM}}$。

预言机只揭示了 M_L，即处理长作业的机器集合，但不揭示长作业的具体身份。这种有限的信息要求我们偏离经典的基于线性规划的近似

框架。我们“猜测” M_L 中的每台机器只处理一个长作业，并构造另一个线性规划，将从预言机$_{\mathsf{LM}}$ 获得的信息纳入其中，以得到关于 C^*_{max} 的更强下界 βT，其中 $\beta \geq 1$。然而，这个线性规划处理了将长作业以及（可能还有）短作业分配到 M_L 中机器上的问题，却无法使我们获得高效的近似。

因此，我们考虑一个较弱的线性规划：在构造机器负载约束时，将分配给每台机器 $i \in M_L$ 的长作业的处理时间视为恰好为 ℓ，尽管其实际处理时间可以是 $(\ell, \beta T]$ 中的任意值。由此得到的线性规划具有一些有用的结构性质：我们证明，任何基本可行解都可以通过构造一个与之关联的二分图并证明该图是伪森林来舍入（类似于 [15]）。

最后，通过分类讨论，我们推导出该方法所达到的近似比是参数 α 的函数，并对其取值进行优化（对于任意 $\beta > 1$ 值）。重要的是，我们的分析还考虑了错误猜测的情况，即认为 M_L 中的一台机器处理多个长作业。

针对 Graph Balancing 的分析。我们的分析基于对 [10]中为 Graph Balancing 问题提出的方法的改进。借助预言机$_{\mathsf{LM}}$，我们通过证明对于不包含“过长”作业的实例，[10] 中确立的 1.75 因子可以得到改进，从而获得更优的近似保证（见第 4 节）。

预言机增强的最大最小公平分配。我们针对最大最小公平分配调整了经典的最大完工时间舍入技术，同时结合从预言机$_{\mathsf{LM}}$获得的信息。我们制定了一个定制的线性规划，将可行分配空间限制在不会与预言机输出相矛盾的范围内，同时确保 M_L 中的机器能够从长作业中获得足够的负载。随后，我们利用伪森林二分图分析对分数解进行舍入，确保每台机器的负载至少达到 $\min\{\alpha, 1-\alpha\}C^*_{min}$，其中 $\alpha \in \in (0,\ 1)$（见第 5 节）。

困难性结果。经典的 Min Makespan [15] 和 GraphBalancing [4] 近似困难性结果确立了$_{3\ 2}$的下界。在其从 3D-匹配和 3-SAT 构造的归约实例中，最优最大完工时间为 $C^*_{max} = 2$，作业长度为 1 或 2，长度为 2 的作业用于阻止机器接收额外作业。这些被阻塞的机器对应于不属于完美匹配的三元组，或对应于被赋予 False 值的文字。如果预言机揭示哪些机器处理长度大于 αC^*_{max}的作业（对于任意 $\alpha \in \in (0,\ 1)$），这种困难性便会从根本上瓦解。为此，我们引入了从 3-SAT 出发的新归约。具体而言，我们设计了新的变量构件，确保相互冲突的文字获得不同的分配，同时保持 $C^*_{max} = 2$，并保证在所构造的调度实例中，至少被分配一个长作业的机器不依赖于具体实例，从而使预言机的输出失去作用（见第 6 节）。

1.3 相关研究

无需预言机的结果。 Min Makespan 问题最早由 Horowitz 和 Sahni [12], 研究，他们提出了一种（$1+\epsilon$）近似算法，其运行时间在机器数量固定时为多项式时间，但随机器数量呈指数增长。Lenstra、Shmoys 和 Tardos [15], 提出了首个适用于任意数量机器的多项式时间近似算法，达到了 2 的近似比。Shchepin 和 Vakhania [17] 随后将这一保证略微提升至 $2-1/m$，其中 m 为机器

的数量。尽管如此，目前尚无已知的、与 2 保持正距离的近似比。论文 [15]还表明，该问题在小于 $3/2$ 的任何因子下进行近似都是 NP困难的。因此，缩小 $3/2$与 2 之间的差距仍然是一个长期悬而未决的开放问题。

Restricted Assignment 问题由 [15], 提出，他们针对 MinMakespan 的 2-近似适用于这一特殊情形。Svensson [18] 随后将配置线性规划的整数间隙界定为 $33/17+\epsilon$，但其结果并未给出构造调度的多项式时间算法。Jansen 和 Rohwedder [13] 将此界改进为 $11/6+\epsilon$，并获得了一个具有相同近似比的准多项式时间算法。因此，目前尚无多项式时间算法能得到严格优于 2 的近似比，而准多项式时间内则可以突破 2 这一界限。该问题在任何小于 $3/2$ [15] 的因子下进行近似都是 NP困难的。

Graph Balancing问题于 [10]提出。该论文针对 Min Makespan [15], 基于更紧的线性规划松弛，给出了一种 1.75-近似算法。他们还证明，在近似该问题时，若近似因子优于 $3/2$，则该问题是 NP困难的。[4]给出了适用于所有边权重均属于 $\{1,k\}$ 的实例的 $3/2$-近似算法。论文 [14] 综述了其他相关结果，并给出了适用于树的全多项式时间近似方案。

Santa Claus问题（最大最小公平分配）是无关并行机调度中另一个得到充分研究的目标。由 Chakrabarty、Chuzhoy 和 Khanna [9], 给出的目前已知最佳多项式时间保证是 $O(n^{\epsilon})$，对每个固定常数 $\epsilon>0$ 成立。他们证明，可以在拟多项式时间内得到多对数近似（另见 [6]）。基于配置线性规划的 $\sqrt{m}\log^3 m$-近似在 [3] 中给出。论文 [2] 提出了一种算法，仅在常数因子范围内近似最优解的值 。我们注意到，本文在可访问预言机$_{\mathsf{LM}}$ 的情况下得到的比值 2，与针对 Santa Claus [7] 的目前已知最佳近似困难性结果 2 相吻合。我们经预言机增强的 2-近似建立在 [7], 所给出的一种算法之上，该算法计算出一种分配，使每台机器被分配的负载至多为 $C^*_{min}-\max_{i,j} p_{ij}$。

可信建议。 可信建议最广泛地应用于在线算法的研究中。在经典的建议复杂度模型中，掌握完整输入序列的预言机会向在线算法提供一个二进制建议串。该建议可以描述输入中未知的部分，也可以编码最优离线解的决策或结构性质。一个核心目标是刻画建议位数量与算法所达到的竞争比之间的权衡关系（例如参见 [11, 8]）。无建议的情形还原为经典在线模型，而足够详细的建议则可以编码最优离线算法的全部决策。

在离线设置中，可信建议受到的关注较少。一个早期的概念先驱是 Arora 的受邀报告 [1], ，其中提出，可以将某些近似算法视为从可信预言机获取关于最优解的少量正确比特的算法。如果可能的建议字符串数量为多项式级，随后便可通过枚举所有此类字符串来消除预言机。据我们所知，以往工作尚未根据基于解的调度预言机所披露的结构信息，对其进行系统比较，也未将这类预言机组织成信息细化层级。我们的目标是利用此类层级，确定哪些形式的正确结构信息足以突破经典近似比和竞争比障碍。

学习增强近似。我们的框架也与迅速发展的学习增强算法研究相关。学习增强算法除了接收问题输入之外，还会接收可能带有噪声的预测，例如根据历史数据或由学习得到的模型生成的预测。针对我们所研究问题的学习增强近似算法直到最近才开始出现。Baba、Bampis 和 Mitropoulos [5] 研究了 Min Makespan，其中涉及对长作业分配的预测。他们的算法在预测准确时可实现 $(1+\epsilon)$-近似（与我们的预热结果类似），而其保证会随预测误差平滑退化，且近似比始终不超过经典近似比 2。Xefteris [19]考虑了 Restricted Assignment，其中预测分配可能不可行，并给出了取决于为达到最优调度而必须重新分配的总处理量的保证，同时仍以最佳无预言机近似保证作为最坏情况界。作为 Restricted Assignment 的一个特例，该框架也适用于 Graph Balancing，但它并未利用其图结构来实现针对问题的改进。据我们所知，目前还没有针对 Graph Balancing或任何 Santa Claus 问题变体的学习增强近似算法。

2 在非相关机器上最小化最大完工时间

本节将展示，通过使用各种预言机来解决 Min Makespan 问题，可以获得更优的近似保证。

2.1 预热：长作业调度预言机

回顾一下，对于给定的实例和阈值 $\ell > 0$，预言机$_{\mathsf{LS}}$ 会观察一个特定的最优调度 OPT，并报告 OPT 中所有处理时间大于 ℓ 的作业的分配。

定理 2.1 预言　机$_{\mathsf{LS}}$ 可用于找到最大完工时间至多为 $C^*_{max}+\ell$ 的调度，其中任意 $\ell > 0$ 均成立。

已移至 APP **证明：** 给定 $\ell > 0$，令 $J_L \subseteq J$ 为 OPT 中处理时间高于 ℓ的作业集合。预言机$_{\mathsf{LS}}$ 报告 $j \in J_L$中的每个作业在 OPT 中被分配到的机器 $m(j)$。考虑图 1 中的线性规划公式 **线性规划**$_{\mathsf{LS}}(t)$，它固定预言机给出的分配，并以最大完工时间为参数。

$$
\begin{aligned}
\mathbf{LP}_{\mathsf{LS}}(t)\quad & \sum_{i\in M} x_{ij} = 1 \quad \forall j \in J && (1)\\
& \sum_{j\in J} p_{ij}x_{ij} \le t, \forall i \in M && (2)\\
& x_{ij} = 1 \quad \forall j \in J_L \wedge i = m(j)\\
& x_{ij} = 0 \quad \forall j \in J_L \wedge i \neq m(j)\\
& x_{ij} = 0 \quad \forall j \in J \setminus J_L \wedge i \in M\text{，其中 } p_{ij} > \ell\\
& x_{ij} \ge 0 \quad \forall j \in J \setminus J_L
\end{aligned}
$$

图 1：线性规划公式 **线性规划**$_{\mathsf{LS}}(t)$

求最小值　　最小整数 T，使得 **线性规划**$_{\mathsf{LS}}(T)$ 有可行解。这可以在多项式　　al

时间内通过二分搜索完成。注意，$T \le C^*_{max}$。求出 **线性规划**$_{\mathsf{LS}}(T)$ 的一个基本解。使用 [15] 的方法对该解进行舍入。由于该解中的任意分数 x_{ij} 都满足作业 j 的长度至多为 ℓ，因此所得调度的最大完工时间至多为 $T + \ell \le C^*_{max} + \ell$。 ■

由此得到的近似比为 $1 + \frac{\ell}{C^*_{max}}$。因此，要实现 $(1+\epsilon)$-近似，我们可以使用 $\ell \le \epsilon T$ 查询预言机 Oracle$_{\mathsf{LS}}$，其中 $T \le C^*_{max}$ 是使得 **线性规划**$_{\mathsf{LS}}(T)$ 存在可行解的最小整数。

2.2 长作业代表预言机

我们考虑预言机 Oracle$_{\mathsf{LR}}$：对于给定的实例和阈值 $\ell > 0$，它会观察一个特定的最优调度 OPT，并针对每台 $i \in M$ 至少处理一个处理时间大于 ℓ 的作业的机器，指出恰好一个此类作业的身份。我们将作业分配由预言机 Oracle$_{\mathsf{LR}}$ 揭示的作业称为代表作业，并将这些作业的集合记为 $J_R \subseteq J$。对于每个 $j \in J_R$，令 $m(j)$ 表示 OPT 中分配给它的机器。

定理 2.2 预言机$_{\mathsf{LR}}$ 可在最小完工时间上实现黄金比例 $\varphi \approx 1.618$-近似。**证明：** 回顾图 2 中给出的 [15]线性规划公式。

$$\textbf{LPLST}(t) \qquad \sum_{i \in M_j[t]} x_{ij} = 1 \text{ 对于 } j = 1, \ldots, n \tag{3}$$

$$\sum_{j \in J_i[t]} p_{ij} x_{ij} \le t, \text{ 对于 } i = 1, \ldots, m \tag{4}$$

$$x_{ij} \ge 0 \quad \text{对于 } j \in J_i\ (t) \quad i = 1, \ldots, m$$

图 2：线性规划公式**LPLST**（t）

在该线性规划中，记号 $J_i[t]$、$t \in \mathbb{Z}^+$ 表示处理时间小于或等于 t 的机器 i 上的作业子集，而记号 $M_j[t]$、$t \in \mathbb{Z}^+$ 表示作业 j 的处理时间小于或等于 t 的机器子集。Lenstra 等人 [15] 找到使**LPLST(*t*)** 可行的最小整数 T，并输出最大完工时间为 $2T$ 的调度。

给定一个 Min Makespan 实例，我们应用 [15] 来求出上文定义的 T。我们以阈值 $\ell = \alpha \cdot T$ 查询预言机 $_{\mathsf{LR}}$，其中 $0 < \alpha < 1$ 的值稍后确定。

我们"猜测"预言机 $_{\mathsf{LR}}$ 揭示的集合 J_R，是在 OPT 中处理时间大于 ℓ 的所有作业组成的集合。假设这一猜测正确，我们便使用上文定义的线性规划公式 **线性规划**$_{\mathsf{LS}}(t)$（其中 J_R 起到 J_L 的作用），尝试找出使 **线性规划**$_{\mathsf{LS}}(T')$ 存在可行解的最小整数 T'。如果不存在这样的 T'，那么我们就有证据表明猜测有误。如果存在这样的 T'，那么如上所述，我们可以得到一个最大完工时间至多为 $T' + \ell$ 的调度。注意，由于 **线性规划**$_{\mathsf{LS}}(t)$ 的任何可行解也是 **LPLST(*t*)** 的可行解，因此 $T' \ge T$。此外，如果我们的猜测正确，那么我们还能保证 T' 不超过 OPT 的最大完工时间，这意味着近似比为 $\frac{T' + \alpha T}{T'} \le 1 + \alpha$。

如果我们的猜测有误，那么 M_R 中至少有一台机器处理时间大于 ℓ的作业不少于两个。这意味着 OPT 的最大完工时间至少为 2ℓ。由于我们已经使用 [15], 获得了最大完工时间为 $2T$ 的有效调度，因此在这种情况下，近似比为 $\frac{2T}{2\alpha T} = \frac{1}{\alpha}$。

我们输出最大完工时间为 $T' + \ell$ 的调度（如果存在）和最大完工时间为 $2T$ 的调度中较好的一个。由于我们无法控制猜测是否正确，因此该调度的近似比为 $\max\{1+\alpha, \frac{1}{\alpha}\}$。最优 α 在 $1+\alpha = \frac{1}{\alpha}$ 时达到，这意味着近似比为 $1+\alpha = \varphi = \frac{1+\sqrt{5}}{2} \approx 1.618$

■

2.3 长作业机器预言机

接下来，我们考虑最弱的预言机 $\mathsf{Oracle}_{\mathsf{LM}}$。它不提供具体的作业到机器分配，但我们将证明，它仍能改进在 [15]中达到的 2 的近似比。对于给定的实例和阈值 $\ell > 0$，预言机 $\mathsf{Oracle}_{\mathsf{LM}}$ 会观察一个具体的最优调度 OPT，并揭示机器子集，记为 M_L，使得每台机器 $i \in M_L$ 在 OPT 中处理至少一个处理时间大于 ℓ 的作业。

给定一个 Min Makespan 的实例，我们再次应用 [15]，以找到使$\mathbf{LPLST}(t)$可行的最小整数 T。我们以阈值 $\ell = \alpha \cdot T$ 向预言机 $\mathsf{Oracle}_{\mathsf{LM}}$ 发起查询，其中 $0 < \alpha < 1$ 将在之后确定，并获得相应的集合 M_L。

与第 2.2 节类似，我们"猜测" M_L 中的每台机器都被分配了恰好一个处理时间大于 ℓ的作业。假设这一猜测成立，我们定义一个新的线性规划公式 $\mathbf{LP}_{\mathsf{LM}}(t)$，如图3所示。

$$
\begin{aligned}
\mathbf{LP}_{\mathsf{LM}}(t) \quad & \sum_{i \in M_j[t]} x_{ij} = 1 \quad \text{for } j = 1, \ldots, n && (5)\\
& \sum_{j \in J_i[\alpha T, t]} x_{ij} = 1 \quad \text{for } i \in M_L && (6)\\
& \sum_{j \in J_i(\alpha T)} p_{ij} x_{ij} \le t \text{ for } i \in M \setminus M_L && (7)\\
& \sum_{j \in J_i[t-\alpha T] \cup J_i[\alpha T, t]} p_{ij} x_{ij} \le t \quad \text{对于 } i \in M_L && (8)\\
& x_{ij} = 0 \quad \text{for } i \in M \setminus M_L,\ j \in J_i[\alpha T, t] && (9)\\
& x_{ij} = 0 \quad \text{for } i \in M_L,\ j \in J_i(t - \alpha T, \alpha T) && (10)\\
& x_{ij} \ge 0 \quad \text{对于 } i \in M,\ \ j \in J_i[t] && (11)
\end{aligned}
$$

图3：线性规划公式 **线性规划**$_{\mathsf{LM}}(t)$

在 **线性规划**$_{\mathsf{LM}}(t)$ 中，记号 $J_i[s,t]$, $s,\ t \in \mathbb{Z}^+$ 表示处理时间位于机器 i 上（闭）区间 $[s,t]$ 内的作业子集。类似地，记号 $J_i(s,\ t)$, $s,\ t \in \mathbb{Z}^+$ 表示处理时间位于机器 i 上（开）区间 $(s,\ t)$ 内的作业子集，而记号 $J_i(t)$, $t \in \mathbb{Z}^+$ 表示处理时间位于机器 i 上（开）区间 $(0,\ t)$ 内的作业子集。

约束（5）确保每个作业都被分配到某台机器上，且其长度至多为 t。约束（6）限制 M_L 中的每台机器恰好处理一个长作业。约束（7）确保每台不处理长作业的机器所处理的短作业总长度至多为 t。约束（8）将处理长作业的每台机器上的总负载限制为 t。由于长作业的长度至少为αT，因此除长作业外，只能处理长度至多为 $t - \alpha T$ 的作业。约束（9）确保 $M \setminus M_L$ 中的每台机器都不会被分配到一个在该机器上具有长处理时间的作业。

约束（10）确保 $M \setminus M_L$ 中的每台机器都不会被分配到一个在该机器上处理时间为 $t \in$ （$t - \alpha T$， αT）的作业。（我们指出，如果 $t \geq 2\alpha T$，则此约束是空泛的。）最后，约束（11）放宽了二元变量 x_{ij}的整数性。

现在，我们求出使**线性规划**$_{\mathsf{LM}}$(t)可行的最小整数 T'。注意，对于任何值 t，**线性规划**$_{\mathsf{LM}}$ 都可能不可行。如果存在可行解，那么很容易看出 $T' \geq T$，因为**线性规划**$_{\mathsf{LM}}$(t)的每个可行解也是**LPLST**(t)的可行解。要说明这一点，请注意，约束 (8) 和 (7) 与约束 (9) 和 (10) 合起来比约束 (4) 更严格，因为约束 (8) 和 (7) 中的求和是在约束 (4) 求和所涉及的相应集合的子集上进行的，而且不属于约束 (8) 和 (7) 所定义子集的变量都固定为零。

定理 2.3 预言机$_{\mathsf{LM}}$ 给出最小完工时间的 $\sqrt{3}$-近似。

证明： 回顾一下， T'是使**线性规划**$_{\mathsf{LM}}$(T')存在可行解的最小整数值。我们分两种情况讨论。

情形 1：线性规划$_{\mathsf{LM}}$(t)没有可行解。这意味着我们的猜测是错误的，并且 OPT 中至少有一台机器$i \in M_L$ 被分配了多个长作业。因此， $C^*_{max} > 2\alpha T$。由于按照 [15] 中的方法对 **LPLST** 进行舍入后得到的调度，其最大完工时间至多为 $2T$，所以近似比至多为 $\frac{1}{\alpha}$。

情形2：线性规划$_{\mathsf{LM}}$(t)有可行解，并且 T' 是使 **线性规划**$_{\mathsf{LM}}$(t)可行的最小整数。在这种情况下， T' 是所有满足以下条件的调度的最大完工时间下界： M_L 中的每台机器都恰好被分配一个长作业。我们根据 T' 的值区分两个子情形。

情形2.1： $T' \geq 2\alpha T$。在这种情况下， $2\alpha T$是任意可行调度的最大完工时间下界，因为它也是所有至少有一台 M_L 中的机器被分配多个长作业的调度的最大完工时间下界；因此，近似比至多为 $\frac{1}{\alpha}$。

情形 2.2： $T' < 2\alpha T$。在这种情况下，我们构造另一个线性规划公式，它略弱于**线性规划**$_{\mathsf{LM}}$，但可以按照 [15] 的技术进行舍入。定义一个新的线性规划，记为**线性规划**$_{\mathsf{LM}}$，如图 4 所示，其中使用了 T、T' 和 α。

$\widehat{\mathbf{LP}}_{\mathsf{LM}}$

$$\sum_{i \in M_j[T']} x_{ij} = 1 \quad \text{for } j = 1, \ldots, n \tag{12}$$

$$\sum_{j \in J_i[\alpha T, T']} x_{ij} = 1 \quad \text{for } i \in M_L \tag{13}$$

$$\sum_{j \in J_i(\alpha T)} p_{ij} x_{ij} \leq T' \text{ for } i \in M \setminus M_L \tag{14}$$

$$\sum_{j \in J_i[T' - \alpha T]} p_{ij} x_{ij} \leq T' - \alpha T \quad \text{对于 } i \in M_L \tag{15}$$

$$x_{ij} = 0 \quad \text{for } i \in M \setminus M_L,\ j \in J_i[\alpha T, T'] \tag{16}$$

$$x_{ij} = 0 \quad \text{对于 } i \in M_L,\ j \in J_i\ (T' - \alpha T,\ \alpha T) \tag{17}$$

$$x_{ij} \geq 0 \quad \text{对于 } i \in M,\ j \in J_i[T'] \tag{18}$$

图 4：线性规划公式**线性规划**$_{\mathsf{LM}}$

约束 (12) 确保每个作业都被分配到某台机器上，且其长度不超过 T'。约束 (13) 规定 M_L 中的每台机器恰好处理一个长作业。约束 (14) 确保每台不处理长作业的机器所处理的短作业总长度至多为

T'。约束 (15) 将短作业对每台处理长作业的机器负载的贡献限制为 $T' - \alpha T$。约束 (16) 确保不会将处理时间较长的作业分配给 $M \setminus M_L$ 中的任何机器。约束 (17) 确保不会将处理时间为 $t \in (T' - \alpha T,\ \alpha T)$ 的作业分配给 $M \setminus M_L$ 中的任何机器。（注意，$T' - \alpha T < \alpha T$，因为 $T' < 2\alpha T$。）最后，约束 (18) 放宽了二元变量 x_{ij} 的整数性。

注意到，**线性规划**$_{\mathsf{LM}}(T')$ 的任一可行解也是**线性规划**$_{\mathsf{LM}}$ 的可行解（反之则不然），因为约束（15）比约束（8）弱。由于**线性规划**$_{\mathsf{LM}}$ 中的约束（16）和（17）将某些变量固定为零，我们从公式中省略这些变量，并忽略这些约束。因此，并且由于 $T' - \alpha T < \alpha T$，**线性规划**$_{\mathsf{LM}}$ 中的每个变量恰好出现在约束（13）、（14）和（15）中的一个约束内。我们将在下面的分析中用到这一事实。

以下结果将在分析我们的算法所采用的舍入子程序时发挥核心作用。

命题 2.4 的任一极点**线性规划**$_{\mathsf{LM}}$ 至多有 $n + m + |M_L|$ 个取严格正值的项。**证明：** 在**线性规划**$_{\mathsf{LM}}$ 中，有 $r = O(mn)$ 个变量以及五类约束：n 个类型为（12）的约束、$|M_L|$ 个类型为（13）的约束、$|M \setminus M_L|$ 个类型为（14）的约束、$|M_L|$ 个类型为（15）的约束，以及 r 个类型为（18）的约束。设 $\bar{\mathbf{x}}$ 是**线性规划**$_{\mathsf{LM}}$的一个极点。则有 r 个线性无关约束在 $\bar{\mathbf{x}}$ 处取等号。在这些约束中，类型为（12）、（13）、（14）和（15）的约束至多有（$n + m + |M_L|$）个。因此，至少有（$r - n - m - |M_L|$）个类型为（18）的约束取等号，即 $\bar{\mathbf{x}}$ 中有（$r - n - m - |M_L|$）个项等于零。因此，取严格正值的项至多有 $n + m + |M_L|$ 个。 ∎

推论 2.5 在**线性规划**$_{\mathsf{LM}}$ 的任意极点中，至多有 $m + |M_L|$ 个作业是分数分配的。**证明：** 设 $\bar{\mathbf{x}}$ 为**线性规划**$_{\mathsf{LM}}$的一个极点，并令 k 表示 $\bar{\mathbf{x}}$中严格为正的项数。根据命题 2.4，$k \le n+m+|M_L|$。注意，k 对应于$\mathbf{x}$ $\bar{}$中作业到机器的分配数。每个作业至少需要一次分配；如果作业 j j 被分数分配，则它需要多于一次分配。因此，$\bar{\mathbf{x}}$ 中被分数分配的作业数至多为 $k - n \le m + |M_L|$。 ∎

设 $\bar{\mathbf{x}}$ 是线性规划$_{\mathsf{LM}}$的一个基本可行解。回顾一下，每个变量 x_{ij} 恰好出现在线性规划$_{\mathsf{LM}}$的两个约束中：一个是指派约束（在 (12) 中），另一个是机器侧约束（在 (13)、(14) 和 (15) 中）。我们利用这一性质构造如下二分图 $G = (J, C, E_G)$。每个顶点 $j \in J$ 对应一个作业。集合 C 对应于 (13)、(14) 和 (15) 中的约束，其中每台机器 $i \in M \setminus M_L$ 对应一个顶点，每台机器 $\hat{}i \in M_L$对应两个顶点，分别记为 i 和 i。E_G 中的每条边对应一个在 $\bar{\mathbf{x}}$ 中严格为正的变量 x_{ij}。如果$i \in M \setminus M_L$ 且作业 $j \in J_i$(αT)，则添加一条边 $(j, i) \in E_G$。类似地，如果 $i \in M_L$ 且作业 $j \in J_i[T' - \alpha T]$，则添加一条边 $(j, \hat{}i) \in E_G$。最后，如果 $i \in M_L$ 且作业 $j \in J_i[\alpha T, T']$，则添加一条边 $(j, \hat{}i) \in E_G$。G 中的边总数等于 $\bar{\mathbf{x}}$ 的支撑集大小，即 $\bar{\mathbf{x}}$ 中严格为正的变量数。因此，G 有 $(n + m + |M_L|)$ 个顶点，并且根据命题 2.4，$|E_G| \le n + m + |M_L|$。

如果一个图是连通的且至多包含一个环，则称其为伪树，这意味着它要么是一棵树，要么是一棵树加上一条额外的边。如果图的每个连通分量都是伪树，则称其为伪森林。

命题 2.6 二分图 $G = (J, C, E_G)$ 是一个伪森林。

证明： 我们区分两种情况 .

情形 1： G 是连通的。因此，G必须包含一个生成树子图。该生成树有$n + m + |M_L| - 1$ 条边。因此，G 是一棵生成树再加上至多一条额外边，也就是一棵伪树。

情形2： G 不是连通的。我们证明，在这种情况下， G 的每个连通分量都是一棵伪树。反设存在一个连通分量 $H=(J_H,\ C_H,\ E_H)$，它不是一棵伪树，其中 $J_H \subseteq J$、 $C_H \subseteq C$ 和 $E_H \subseteq E_G$。将图G的其余部分记为 $H=(J_{\bar{H}},\ C_{\bar{H}},\ E_{\bar{H}})$；也就是说， $J_{\bar{H}}=J\setminus J_H$、 $C_{\bar{H}}=C\setminus C_H$ 和 $E_{\bar{H}}=E_G\setminus E_H$。

现在，我们使用 H 和 H 来定义两个受限线性规划。第一个线性规划记为$\mathbf{LP}_H$，由所有变量 x_{ij} 组成，这些变量对应的边属于 E_H。$\mathbf{LP}_H$ 的约束通过将$\mathbf{LP}_{\mathsf{LM}}$ 的约束限制到$\mathbf{LP}_H$的变量而得出。具体而言，$\mathbf{LP}_H$ 的约束包括：式（12）中由 J_H 中的作业索引的$\mathbf{LP}_{\mathsf{LM}}$ 约束，且这些约束仅涉及对应边属于 E_H的变量；以及式（13）、（14）和（15）中由 C_H 索引的约束，且这些约束仅涉及对应边属于 E_H的变量。第二个线性规划记为$\mathbf{LP}_{\bar{H}}$，其定义类似，涉及所有 x_{ij} 对应边属于 $E_{\bar{H}}$ 的变量。具体而言，$\mathbf{LP}_{\bar{H}}$ 的约束包括：式（12）中由 $J_{\bar{H}}$ 中的作业索引的$\mathbf{LP}_{\mathsf{LM}}$ 约束，且这些约束仅涉及对应边属于 $E_{\bar{H}}$的变量；以及式（13）、（14）和（15）中由 $C_{\bar{H}}$ 索引的约束，且这些约束仅涉及对应边属于 $E_{\bar{H}}$的变量。

我们声称， $\bar{\mathbf{x}}$ 中严格为正的项可以拆分为两个向量： $\bar{\mathbf{x}}_H$ 和 $\bar{\mathbf{x}}_{\bar{H}}$，它们分别是下**线性规划**$_H$ 和**线性规划** $_{\bar{H}}$的可行解。事实上，对于任意一个正项 x_{ij}，相应的边要么属于 E_H，要么属于 $E_{\bar{H}}$。此外，由于 H 和 H 是不同的连通分量， x_{ij} 所参与的两个约束都由同一个连通分量中的顶点索引。因此，对应于 E_H 的变量只出现在由 $J_H\cup C_H$索引的约束中，而对应于 $E_{\bar{H}}$ 的变量只出现在由 $J_{\bar{H}}\cup C_{\bar{H}}$索引的约束中。由于 $\bar{\mathbf{x}}$ 是**线性规划**的可行解，因此 $\bar{\mathbf{x}}_H$ 是**线性规划**$_H$ 的可行解，而 $\bar{\mathbf{x}}_{\bar{H}}$ 是**线性规划** $_{\bar{H}}$的可行解。

接下来，我们证明 $\bar{\mathbf{x}}_H$ 不可能是$\mathbf{LP}_H$的一个极点。由于 H 是连通的但不是伪树，因此至少包含两个环。因此， $|E_H|>|J_H|+|C_H|$。现在考虑$\mathbf{LP}_H$的约束矩阵。注意，每个变量 x_{ij} 都与约束矩阵中的一列相对应。根据 H的定义，与 E_H 中的变量对应的约束矩阵列，仅在由 $J_H\cup C_H$索引的行中具有非零项。因此，这些列所处空间的维数至多为$|J_H|+|C_H|$。由于与$\mathbf{LP}_H$相关联的列数满足 $|E_H|>|J_H|+|C_H|$，这些列是线性相关的。回顾一下，线性规划的任何基本可行解都对应于一组线性无关的列。因此， $\bar{\mathbf{x}}_H$ 不是$\mathbf{LP}_H$的基本解。

因此， $\bar{\mathbf{x}}_H$ 可以写成其他可行点的凸组合；也就是说， $\bar{\mathbf{x}}_H=\lambda\bar{\mathbf{y}}_1+(1-\lambda)\bar{\mathbf{y}}_2$,其中 $\bar{\mathbf{y}}_1\neq\bar{\mathbf{y}}_2$。那么， $\bar{\mathbf{x}}$ 也可以写成原始实例的可行点的凸组合，即 $\bar{\mathbf{x}}=\lambda(\bar{\mathbf{x}}_H+\bar{\mathbf{y}}_1)+(1-\lambda)(\bar{\mathbf{x}}_H+\bar{\mathbf{y}}_2)$。这与 $\bar{\mathbf{x}}$ 是一个极点矛盾。因此，G是一个伪森林。

■

现在我们说明如何通过对解 $\bar{\mathbf{x}}$进行舍入来获得一个可行调度。首先，对于 $\bar{\mathbf{x}}$ 中的每个变量$x_{ij}=1$，我们将作业 j 分配给机器 i。此外，我们从 J 中移除顶点 j，并移除与j 相连的唯一一条边 E_G。如果所得子图在 C 中有孤立顶点，我们也将其移除。

所得子图 $G'=(J',\ C',\ E'_G)$ 仍然是一个伪森林。此外， J' 中的每个顶点的度都至少为二。这是因为每条边都对应于一个严格小于 1 的变量 x_{ij}，而每个作业都需要被完整分配。

我们通过寻找一个匹配，将 J' 中的所有顶点与 C' 中的顶点匹配起来，并将作业 j 分配给与 $j\in J'$ 匹配的机器，从而把 J' 中对应于顶点的每个作业分配给一台机器。由此可知，这些作业分别被分配给不同的机器。

匹配的求法如下。假设存在一个度为 1 的顶点 $i \in C'$，并令 (j, i) 为 E'_G 中与顶点 i 相接的唯一一条边。我们将 j 与 i 匹配。此外，我们从 J' 中移除顶点 j，并从 E'_G 中移除所有与 j 相接的边。我们还从所得子图 C' 中移除 i 以及其他任何孤立顶点。我们持续这一过程，直到 C' 中不再有度为 1 的顶点。

此时，C' 中的所有顶点的度都大于 1。由于 J' 中的所有顶点的度也都大于 1，且 G' 是一个伪森林，因此 G' 必定由若干个互不相交的（偶）环组成，所以G' 存在完美匹配。

接下来只需分析所得舍入解的质量。注意，对于每台机器 $i \in M \setminus M_L$，至多分配给它一个满足 $x_{ij} < 1$的作业。因此，机器 i 的最大完工时间不超过$T' + \alpha T$。类似地，每台机器 $i \in M_L$至多分配给它一个作业 j，该作业满足：(i) 作业 j在机器 i上的处理时间至多为 $T' - \alpha T$；(ii) $x_{ij} < 1$。因此，机器 i 上所有处理时间至多为 $T' - \alpha T$的已分配作业的总处理时间不超过 $2(T' - \alpha T)$。

我们仍需处理 M_L上的长作业。考虑一台机器 $i \in M_L$。我们声称，机器 i 至多被分配一个来自 $J_i[\alpha T, T']$的作业。假设 j 是这样一个作业。如果 $x_{ij} = 1$，那么在将作业 j 分配给 i后，C 中与该约束对应的顶点会变为孤立顶点，并从 G' 中移除，因此不会再有其他长作业分配给机器 i。如果 $x_{ij} < 1$，那么不存在这样的长作业 j'，使得 $x_{i,\ j'} = 1$（因为在这种情况下，i 不会属于 C'）。由于 J' 中的每个作业都与 C' 中不同的顶点匹配，因此不会有其他作业 $j' \in J_i[\alpha T, T']$（使得 $x_{i,\ j'} < 1$）被分配给机器 i。回想一下，作业 j 在机器 i 上的处理时间至多为 T'。结合上述分析可知，机器 i 的最大完工时间不超过 $2(T' - \alpha T) + T' = 3T' - 2\alpha T$。

为了分析情形 2.1 和 2.2 中的近似比，注意，从满足每台机器恰好分配一个长作业的调度中选出的最优解，其最大完工时间至少为 T'。取将原始 $\mathbf{LPLST}(T)$ 舍入后得到的调度与将 $\mathbf{LP}_{\mathsf{LM}}$ 的一个基本解舍入后得到的调度中最大完工时间较小者，近似解的最大完工时间为 $\min\{2T, \max\{T' + \alpha T, 3T' - 2\alpha T\}\}$，其中 $T' \geq T$。代入 $T' = \beta T$，其中某个 $\beta > 1$。结合情形 1、情形 2.1 和情形 2.2 中的界，我们得到近似比有如下上界

$$R = \max\{\frac{1}{\alpha}, \min\{\frac{2}{\beta}, \max\{1 + \frac{\alpha}{\beta}, 3 - 2\frac{\alpha}{\beta}\}\}\},$$

对于任意 $\beta \geq 1$，同时我们可以优化 $\alpha \in (0,1)$。根据标准分析，最优值为 $\alpha = \sqrt{3} - 1$，因此得到 $R \leq \sqrt{3}$。 ∎

3 受限分配

回想一下，RESTRICTED ASSIGNMENT 是 MIN MAKESPAN 的特例，其中 $p_{ij} \in \{p_j, \infty\}$。我们证明，在这个特例中，定理 2.3 中的界可以改进。

定理 3.1 预言机$_{\mathsf{LM}}$ 可为受限分配的最小最大完工时间提供 1。686-近似。**证明：** 注意，由于 $\mathbf{LPLST}(T)$ 有一个可行解，因此 $p_j \leq T, \forall j = 1, \ldots, n$。我们聚焦于定理 2.3 的证明中的情形 2.2（**线性规划**$_{\mathsf{LM}}(T')$ 有一个可行解且 $T' < 2\alpha T$）。和之前一样，任意机器 $i \in M \setminus M_L$ 的最大完工时间都不超过 $T' + \alpha T$。对于机器 $i \in M_L$，我们有，分配给该机器且处理时间不超过 $T' - \alpha T$ 的所有作业在机器 i 上的总处理时间

不超过 2($T' - \alpha T$)。现在，考虑分配给机器 $i \in M_L$ 的长作业。由于该作业的处理时间至多为 T，因此这台机器的完工时间至多为 2($T' - \alpha T$) $+T$。于是，近似比不超过

$$R = \max\{\frac{1}{\alpha}, \min\{\frac{2}{\beta}, \max\{1+\frac{\alpha}{\beta}, 2+\frac{1}{\beta}-\frac{2\alpha}{\beta}\}\}\},$$

对于任意 $\beta \geq 1$，我们可以优化 $\alpha \in$ (0，1)。固定 $\alpha = \frac{\sqrt{33}-3}{4}$ 对于该固定值，函数$f(\beta) = \min\{\frac{2}{\beta}, \max\{1+\frac{\alpha}{\beta}, 2+\frac{1}{\beta}-\frac{2\alpha}{\beta}\}\}$ 的最大值为 $1+\alpha = \frac{\sqrt{33}+1}{4} \approx 1.686$。

■

4 图平衡

现在我们考虑 Graph Balancing问题。回顾一下，该问题是 restrictedassignment 的一个特例，其中每个作业最多可以分配给两台机器。

定理 4.1 预言机$_{\mathsf{LR}}$ 可为图均衡问题提供 $\frac{1}{2}+\frac{3}{4}\sqrt{2} \approx 1.561$-近似。**证明：** 证明沿用了定理 2.2 的证明，同时利用了针对 Graph Balancing 给出的 1.75 近似比 [10]。将这一改进后的近似比代替一般情形下的 2-近似比，我们得到预言机$_{\mathsf{LR}}$ 的近似比为 $\max\{1+\alpha, \frac{1.75}{2\alpha}\}$。最优 α 在 $1+\alpha = \frac{1.75}{2\alpha}$ 时取得，且 $1+\alpha = \frac{1}{2}+\frac{3}{4}\sqrt{2} \approx 1.561$

■

Theo第 4.2 节 预言机$_{\mathsf{LM}}$ 可为图均衡问题提供 $\frac{5}{8}\sqrt{7} \approx 1.6536$-近似 .

证明： 定义 **LP3** 的一个参数化版本，记为 [10], **LPEKS(*t*)**，如图5所示。按照 [10], ，令 $E^B \subseteq E$ 表示权重严格大于 $t/2$的边集。 E^B 中的边称为大边。对于满足 $v \in e$的边 $e \in E$，令 $x_{ev} \in \{0,1\}$ 为将 e 定向至 v的指示变量；也就是说，若 e 定向至 v，则 $x_{ev} = 1$，否则为 $x_{ev} = 0$。在线性规划松弛中，$x_{ev} \geq 0$。

$$\begin{aligned}
\textbf{LPEKS}(t) \quad & x_{ev} + x_{eu} = 1 \quad \forall e = (u,v) \in E && (19)\\
& \sum_{e: v \in e} x_{ev} p_e \leq t, \ \forall v \in V && (20)\\
& \sum_{e \in E^B:\ v \in e} x_{ev} \leq 1, \quad \forall v \in V && (21)\\
& x_{ev} \geq 0 \quad \forall e, v: \ v \in e
\end{aligned}$$

图5：线性规划公式 **LPEKS(*t*)**

求出使 **LPEKS(*t*)** 可行的最小值 t。根据 [10]，可以计算出一个最大负载为 $1.75t$ 的解。注意，由于 t 只是 C^*_{max} 的下界，最优解 OPT 中很可能有两条边从 E^B 指向同一个顶点（如果 t 是 C^*_{max} 的上界，则不会发生这种情况）。

固定 $r > 1$ （稍后计算），假设 rt 是 C^*_{max}的一个上界。请注意，如果这一假设不成立，那么使用 [10] 中的算法计算出的 $1.75t$ 解将得到一个 $1.75/r$ 近似。

从现在起（直到我们确定 r的值为止），我们假设 rt 确实是最优解值的一个上界。

我们让预言机指定一组顶点（机器），这些顶点有一条权重至少为 hrt 的入边（作业），其中 $h > \frac{1}{2}$ 将在稍后指定。我们将所有权重至少为 hrt 的边称为超大边；用 E^H 表示这组边。我们将权重大于 $\frac{1}{2}rt$ 的边称为大边，并用 E^L 表示这组边。注意，$E^H \subseteq E^L$。

令 V^H 为预言机给出的集合。如果一条超大边 $e \in E^H$ 只有一个端点 $v \in V^H$，则它必须定向至 v。我们固定该定向，从实例中移除 e，并将 p_e 加到 v 的权重 q_v 上（初始值为 $q_v = 0$）。我们重复此过程，直到移除所有在 V^H 中只有一个端点的超大边。

现在，我们继续按如下方式对边定向。只要存在一条大边 $e = (u, v) \in E^L$，且其某个端点（不妨设为 u）已经有一条固定入大边，我们就将 e 定向至 v，从实例中移除 e，并将 p_e 加到 q_v。如果 v 也已经有一条固定入大边，或者 $v \in V^H$ 但 $e \notin E^H$，或者 $e \in E^H$ 但 $v \notin V^H$，那么 rt 就不是 C^*_{max} 的上界。

令 $\hat{E}$ 为剩余边集合，令 $\hat{E}^L$ 和 $\hat{E}^H$ 分别为剩余的大边和超大边。此外，令 $\hat{V}^H \subseteq V^H$ 为没有固定入边的顶点集合。注意，每条边 $e \in \hat{E}^H$ 的两个端点都在 $\hat{V}^H$ 和 $|\hat{E}^H| = |\hat{V}^H|$中。因此，$\hat{G}^H(\hat{V}^H, \hat{E}^H)$ 的每个连通分量都是一棵伪树。此外，由于 $\hat{G}^H$ 没有度为 1 的顶点，每个连通分量都是一个环，而 $\hat{G}^H$ 是一组互不相交的环。注意，$V \setminus \hat{V}^H$ 中没有顶点能够有一条满足$p_e > rt - q_v$的入边 $e \in \hat{E}$。此外，$\hat{V}^H$ 中也没有顶点能够有一条满足 $p_e > rt - hrt$ 的入边 $e \in \hat{E} \setminus \hat{E}^H$。

我们定义如下线性规划公式**LPEKS**，见图6。在该线性规划中，除了为每个边端点设置一个变量外，对于$\hat{G}^H$ 中的每个环 c 还设置一个变量 y_c，用于标识该环的全局方向；例如，$y_c = 1$ 为顺时针方向，$y_c = 0$为逆时针方向。对于边$e = (u, v) \in \hat{E}^H$，我们用 $c(e)$表示其所属的环，并假设在 c_e的顺时针方向上，u（成对顶点中的第一个顶点）先于 v（成对顶点中的第二个顶点）。

$$
\begin{aligned}
\widehat{\textbf{LPEKS}} \quad & x_{ev} + x_{eu} = 1 \quad \forall e = (u, v) \in \hat{E} \setminus \hat{E}^H && (22)\\
& q_v + \sum_{e \in \hat{E} : v \in e} x_{ev} p_e \le rt, \ \forall v \in V \setminus \hat{V}^H && (23)\\
& x_{ev} = 0 \quad \text{if } p_e > rt - q_v \quad \forall v \in V \setminus \hat{V}^H \quad \forall e \in \hat{E} : v \in e && (24)\\
& \sum_{e \in \hat{E} \setminus \hat{E}^H : v \in e} x_{ev} p_e + \sum_{e=(u,v) \in \hat{E}^H} y_{c(e)} p_e + \sum_{e=(v,u) \in \hat{E}^H} (1 - y_{c(e)}) p_e \le rt, \ \forall v \in \hat{V}^H && (25)\\
& x_{ev} = 0 \quad \text{if } p_e > rt(1-h) \quad \forall v \in \hat{V}^H \quad \forall e \in \hat{E} \setminus \hat{E}^H : v \in e && (26)\\
& \sum_{e \in \hat{E}^L : v \in e} x_{ev} \le 1, \ \forall v \in V && (27)\\
& x_{ev} \ge 0 \quad \forall e \in \hat{E} : v \in e \\
& 0 \le y_c \le 1 \quad \text{for all cycles } c \text{ in } \hat{G}^H
\end{aligned}
$$

图6：线性规划公式**LPEKS**

约束（22）确保所有剩余边都已定向。约束（23）保证每个顶点 $v \notin \hat{V}^H$ 上的总负载至多为 rt，这是我们对 C^*_{max}施加的上界。约束（24）禁止

在 $p_e > rt - q_v$时将边 e 定向至端点 v。约束（25）规定了顶点全部属于 $\hat{V}^H$的环的定向。它们确保每个这样的环都被赋予单一方向，即顺时针或逆时针，并且其每个顶点上的总负载都不超过 rt。约束（26）涉及顶点 $v \in \hat{V}^H$。每个这样的顶点 v 由于所在环上有一条指向它的超大边，负载已经至少为 rth。因此，可以分配给 v 的额外负载至多为 $rt(1-h)$。最后，约束（27）确保每个顶点至多有一条大边定向指向它。

接下来，我们对**LPEKS**的分数解进行如下舍入。首先，我们对每个变量 y_c进行舍入；也就是说，如果 $y_c \geq 1/2$，我们就按顺时针方向定向该环；否则，就按逆时针方向定向。这处理了 $\hat{E}^H$中的所有边。由 $\hat{E}^H$中的边导致的舍入误差至多为$\frac{1}{2}(rt - hrt)$。对于其余的边，我们采用对 [10]的舍入技术进行修改后得到的方法。注意，尽管最优解的上界为 rt，但我们没有任何权重大于 hrt的边。如下方引理 4.3 所示，在这种情况下，对 [10] 的算法进行修改即可保证近似比为 $\frac{5}{4} + \frac{h}{2}$。由于 [10], 的算法至多将一条进入某个顶点的边向上舍入，因此，$\hat{V}^H$ 中顶点因不在 $\hat{E}^H$ 中的入边而产生的额外舍入误差土界为 $(rt - hrt)$。

总而言之，假设 rt 和 t 分别是 C^*_{max} 的上界和下界，我们得到的近似比为 $\max\{\frac{5}{2}r - \frac{3}{2}hr, \frac{5}{4}r + \frac{1}{2}rh\}$。事实上，左边的项给出了 $\hat{V}^H$ 中顶点的总负载相对于 t 的下界，即 $\left[rt + \frac{1}{2}(rt - hrt) + rt - hrt\right] \cdot \frac{1}{t} = \frac{5}{2}r - \frac{3}{2}hr$。右边的项则由引理 4.3 中的界推出，该引理应用于顶点 $v \notin \hat{V}^H$。由于假设 rt 是最优解的上界可能不成立，因此我们还需要纳入 $\frac{7}{4r}$ 这一项；若该假设不成立，它就是对应的近似比。为了优化近似比，我们需要求 $r \geq 1$ 和 $\frac{1}{2} \leq h \leq 1$ 上 $\max\{\frac{7}{4r}, \frac{5}{2}r - \frac{3}{2}hr, \frac{5}{4}r + \frac{1}{2}rh\}$ 的最小值。首先令第二项和第三项相等，以求得最优的 $h = \frac{5}{8}$。然后令第一项与第二项、第三项相等，得到 $r = \frac{2\sqrt{7}}{5}$，以及近似比 $\frac{5\sqrt{7}}{8} \approx 1.6536$。 ■

现在我们说明，如何修改 [10] 的算法，以便在实例中不包含相对于 C^*_{max} 而言非常大的权重时，获得更优的近似因子；这是定理 4.2 证明所要求的。**引理 4.3** 对于给定的图均衡问题实例 $G(V, E, p)$ 以及给定的 $t > 0$，假设 **LPEKS**(t) 是可行的。如果 $\forall e \in E, \quad p_e < ht$，对于某个 $h \in (\frac{1}{2}, 1]$，则存在一个多项式$\left(\frac{5}{4} + \frac{h}{2}\right)$-近似算法可用于该实例。

证明： 我们沿用 [10] 给出的算法，但不再是在将大边分配给叶节点会使负载增加 $\frac{3}{4}t$ 时才这样分配，而是固定某个 $s \in (\frac{1}{2}, 1]$，并将增加阈值设为 sht。最优的 s 可通过令负载的两种可能增量相等来确定；即 $sht = \frac{1}{2}t + (1-s)ht$。由此可得 $s = \frac{1}{2} + \frac{1}{4h}$，而负载增量为 $\frac{h}{2} + \frac{1}{4}$。 ■

5 圣诞老人问题

本节将介绍我们针对 Santa Claus 问题所得的结果。

5.1 预热 ：使用长作业调度预言机的最大–最小公平分配 e

假设我们有一个预言机，它能观察给定实例的最优分配。令 FOPT 为一个最优解，并令 C^*_{min} 为 FOPT 的值。对于参数 $\alpha \in (0,1)$，令 $J_L \subseteq J$ 为最优分配中处理时间至少为 $\alpha \cdot C^*_{min}$ 的作业子集。强预言机会揭示 J_L，并且对于每个$j \in J_L$，还会揭示在 FOPT 中分配给 j 的机器 $i(j)$。（请注意，预言机不会揭示 C^*_{min}。）

定理 5.1 预言机$_{\mathsf{LS}}$ 可为最大–最小公平分配问题提供 $\frac{1}{1-\alpha}$-近似。

证明： 令 M_L 表示被分配了 J_L 中的作业的机器集合。令 $J(i)$ 表示分配到机器 $i \in M_L$上的长作业集合$\widehat{\ }$。考虑如下定义的线性规划 **SCLP** 。

$$
\begin{aligned}
\widehat{\textbf{SCLP}} \quad & \text{最大化} \quad \omega \\
& \text{s.t.} \quad \sum_i x_{ij} \le 1 \quad \forall j \in J && (28)\\
& x_{i(j),j} = 1 \quad \forall j \in J_L && (29)\\
& x_{i,j} = 0 \quad \forall j \in J \setminus J_L \text{ for which } p_{ij} \ge \alpha \cdot \omega && (30)\\
& \sum_{j \in J \setminus J_L} p_{ij} x_{ij} \ge \omega - \sum_{j \in J(i)} p_{ij} \quad \forall i \in M_L && (31)\\
& \sum_{j \in J \setminus J_L} p_{ij} x_{ij} \ge \omega \quad \forall i \in M \setminus M_L && (32)\\
& 0 \le x_{ij} \quad \forall i \in M,\ j \in J \setminus J_L
\end{aligned}
$$

$\widehat{\ }$图7：线性规划公式 **SCLP**

约束（31）和（32）保证每台机器分配到的总负载至少为 ω。对于 M_L中的机器，处理时间小于 $\alpha \cdot \omega$ 的作业应完成 ω 由 J中的作业（i）产生的负载；而对于不在 M_L中的机器，全部负载都来自不在 J_L 中的作业。约束（30）保证不在 J_L 中的作业被分配到其处理时间小于 $\alpha \cdot \omega$的机器上。

令 ω^* 为以下问题的最优解值：**SCLP**。不难看出，FOPT 导出了**SCLP**的$\widehat{\ }$可行解。因此，$\omega^* \ge C^*_{min}$。此外，[7] 中的定理7所描述的舍入方法$\widehat{\ }$可以应用于**SCLP**。由此可知，该线性规划的任何基本可行解都可以舍入为整数解，使得机器的负载最多减少分数分配作业中处理时间最大的作业的处理时间。

由于所有非整数变量都对应于满足 $p_{ij} < \alpha \cdot \omega$ 的作业和机器，因此舍入过程保证最小效用至少为：

$$\omega^* - \max_{ij:0<x_{ij}<1} p_{ij} > \omega^* - \alpha \cdot \omega^* \ge C^*_{min} - \alpha \cdot C^*_{min} = (1-\alpha) C^*_{min}.$$

■

5.2 使用长作业机器预言机的最大-最小公平分配

假设我们有一个预言机，它能观察给定实例的最优分配。令 FOPT 为一个最优解，C^*_{min} 为 FOPT 的值。对于参数 $\alpha \in (0,\ 1)$，令 $M_L \subseteq M$ 为以下机器子集：

在 FOPT 中执行处理时间至少为 αC^*_{min} 的作业的机器。预言机$_{\mathsf{LM}}$ 会揭示集合 M_L。我们提出一种算法，使用预言机$_{\mathsf{LM}}$来获得 $\max(\frac{1}{\alpha}, \frac{1}{1-\alpha})$-近似。注意，对于 $\alpha = \frac{1}{2}$，这是一个 2-近似。

定理 5.2 预言机$_{\mathsf{LM}}$为最大–最小公平分配问题提供了$\max(\frac{1}{\alpha}, \frac{1}{1-\alpha})$-近似。**证明：** 考虑线性规划 **SCLP**。

$$
\begin{aligned}
\widetilde{\textbf{SCLP}} \quad & \text{最大值} \quad \omega \\
& \text{s.t.} \quad \sum_i x_{ij} = 1 \quad \forall j \in J && (33)\\
& x_{ij} = 0 \quad \forall i \in M \setminus M_L,\, p_{ij} \geq \alpha\omega && (34)\\
& x_{ij} = 0 \quad \forall i \in M_L,\, p_{ij} < \alpha\omega && (35)\\
& \sum_j p_{ij} x_{ij} \geq \omega \quad \forall i \in M \setminus M_L && (36)\\
& \sum_{j: p_{ij} \geq \alpha\omega} x_{ij} = 1 \quad \forall i \in M_L && (37)\\
& 0 \leq x_{ij} \quad \forall i \in M,\, j \in J
\end{aligned}
$$

图 8：线性规划公式 **SCLP**

约束（34）保证，分配给不属于 M_L 的机器的作业处理时间小于 $\alpha\omega$。约束（36）保证，每台不属于 M_L 的机器所分配的总负载至少为ω。根据约束（34），该负载由处理时间小于 $\alpha\omega$的作业产生。约束（37）保证，每台属于 M_L 的机器恰好分配一个处理时间至少为 $\alpha\omega$的作业。根据约束（35），这是该机器上的唯一作业。

令 ω^* 为 $\widetilde{\textbf{SCLP}}$ 的最优解的值。注意，最优分配 FOPT 蕴含 $\widetilde{\textbf{SCLP}}$ 的一个可行解。这是因为每台机器 $i \notin M_L$ 只分配处理时间小于 αC^*_{min} 的作业，且这些处理时间之和必须至少为 C^*_{min}（约束 (36)）；而每台机器 $i \in M_L$ 至少分配一个处理时间为 $\geq \alpha C^*_{min}$ 的作业（约束 (37)）。因此，$\omega^* \geq C^*_{min}$。

我们将 $\widetilde{\textbf{SCLP}}$ 使用 [15] 的技术对 SCLP 的分数解进行舍入。首先，固定 SCLP 的解中每个取整数值的变量 $\tilde{x}_{ij}$。接下来，考虑二分图 $\tilde{G}(\tilde{J}, \tilde{M}, \tilde{E})$，其中 $\tilde{J}$ 是所有满足存在 $i \in M$ 使得其在 SCLP 的解中为分数值的作业 j 的集合，$\tilde{M}$ 是所有满足存在 $j \in J$ 使得其在 SCLP 的解中为分数值的机器 i 的集合，且当且仅当 x_{ij} 为分数值时，$(i, j) \in \tilde{E}$。与 [15] 中一样，$\tilde{G}$ 是一个伪森林。此外，可以应用 [7] 中的引理8，找到一个子集 $S \subseteq \tilde{E}$，使得每个 $j \in \tilde{J}$ 都是 S 中恰好一条边的端点，而每个 $i \in \tilde{M}$ 都是 S 中至少 $deg(i) - 1$ 条边的端点（其中 $deg(i)$ 是 $\tilde{G}$ 中 i 的度）。集合 S 与被固定为 1 的变量共同构成一个作业分配。我们分析每台机器的负载。回想一下，$\omega^* \geq C^*_{min}$：

情形 1： 机器 $i \in M_L$。如果存在 $j \in J$ 使得 $x_{ij} = 1$，那么必然有 $p_{ij} \geq \alpha\omega^* \geq \alpha C^*_{min}$。否则，$i \in \tilde{M}$，并且 $deg(i) \geq 2$，因为约束（37）成立。因此，至少有一条与 i 相接的边，即 (i, j')，属于 S。由于 $p'_{ij} \geq \alpha C^*_{min}$，其负载也至少为 αC^*_{min}。

情形2： 机器 $i \in M \setminus M_L$。设 $J(i) \subseteq J$ 表示在解中分配给 i的作业集合，这些作业可以是分数分配的，也可以是整数分配的。注意，对于每个 $j \in J(i)$，$p_{ij} < \alpha\omega^*$。此外，$\sum_{j \in J(i)} p_{ij} \geq \omega^*$

根据约束 (36)。由于所有满足 $x_{ij} = 1$ 的作业 $j \in J(i)$ 都分配给了 i，并且所有满足 $x_{ij} < 1$ 的作业 $j \in J(i)$ 中，除至多一个作业外，也都分配给了 i，因此机器 i 的负载至少为 $(1-\alpha)\omega^* \geq (1-\alpha)C^*_{min}$。

我们得出结论：每台机器上的负载至少为 $\min(\alpha,\ 1-\alpha)C^*_{min}$。 ■

6 具有预言机访问的近似困难性

尽管本文引入的预言机非常有益，使我们能够绕过经典瓶颈并改进已知的最佳近似比，但其能力本质上是有限的。本节将探讨这些理论边界。我们证明，即使算法得到了这些预言机所提供的信息，底层的调度和分配问题不仅仍然是 NP困难的，而且在近似意义下也是严格困难的。

我们首先提及一个简单直接的结果，涉及本文尚未考虑的预言机。证明基于已知的受限调度近似困难性，以及加入少量虚拟作业。

定理6.1 即使使用一种预言机，该预言机能够揭示 C^*_{max}、实际作业的长度，以及在某个最优解中分配给每台机器的作业数量，对 $R||C_{max}$ 的最大完工时间进行近似，使其近似比小于3 2，仍然是 *NP*困难的。

接下来，我们将证明，只要预言机揭示权重超过 αC^*_{max} 的边的信息，对于任意 $\alpha \leq \frac{1}{2}$，即使可以访问预言机 LM，Graph Balancing 仍然是 NP困难且难以近似的。由于 Graph Balancing 是 $R||C_{max}$ 的一个受限类别，我们的结果显然也涵盖了 Min Makespan 问题。

定理 6.2 对于任意 $\alpha \leq \frac{1}{2}$，在 Graph Bal-ancing 问题中，即使有一个预言机揭示 C^*_{max} 的值以及在某个最优解中具有权重超过 αC^*_{max} 的入边的顶点集合，将最小完工时间近似到小于 3 2 的因子仍然是 *NP*困难的。

证明： 首先假设 $\alpha = \frac{1}{2}$。我们的归约来自 3-SAT。给定一个 3-SAT 公式，我们构造一个 Graph Balancing 实例，使得当且仅当存在最大负载为 $C^*_{max} = 2$ 的定向时，该公式存在满足赋值。此外，接收权重超过 1 的边的顶点集合与解无关。

令 $0 < \epsilon < \frac{1}{3}$。对于每个子句 $C = (\ell_1 \vee \ell_2 \vee \ell_3)$，我们创建一个单独的**子句顶点** v_C。它有 3 条关联边，分别连接到相应的变量构件，每条边代表一个文字 ℓ_i。连接子句顶点和变量构件的所有边的权重均为 $1-\epsilon$。

对于每个变量 x，令 $k = \max(\text{count}(x), \text{count}(\bar{x}))$ 表示文字 x 或其否定 $\bar{x}$在公式中出现的次数的最大值。 x 的**变量构件** 由一个长度为$2k$的环组成。图9展示了一个 $k = 3$的示例。

- 这些顶 点沿着环交替排列，取决于它们承载的文字： v^1_x， $v^1_{\bar{x}}$, v^2_x， $v^2_{\bar{x}}$， ....., v^k_x， v $^k_{\bar{x}}$。
- 环边的权重在 2 和 $1+\epsilon$之间交替。它们的排列方式使得沿顺时针方向移动时，权重为 $1+\epsilon$ 的边总是进入一个 x-节点，而权重为 2 的边总是进入一个 $\bar{x}$-节点。

• 公式中的每个文字出现都会贡献一条权重为 $1-\epsilon$ 的边，连接环中一个类型相符且可用的顶点与对应的子句顶点。

• 请注意，如果文字出现次数不相等（例如，count(x) $=2$ 和 count($\bar{x}$) $=3$，即 $k=3$），我们仍会构建一个包含 $2k$ 个节点、节点类型交替的完整环。 $|\text{count}(x)-\text{count}(\bar{x})|$ 对应出现次数较少的文字的顶点不会与子句顶点相连。

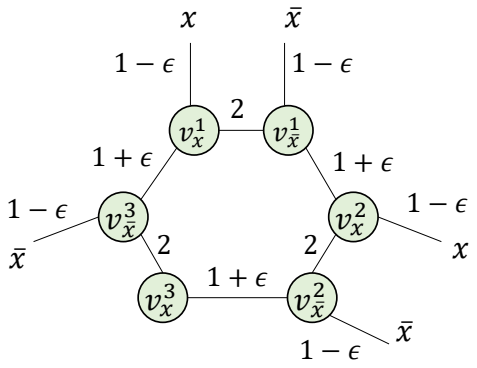

图9：3-SAT 公式中变量 x 的变量构件，其中 x 出现 2 次， $\bar{x}$ 出现 3 次。五个输出端分别与对应的子句顶点相连。

命题6.3 一个 3-*SAT* 公式可满足，当且仅当对应的图平衡实例存在一种定向，使得最大负载为 2。

证明： 假设存在一个满足公式的真值赋值。我们按如下方式对图进行定向 ：

1. 对于每个子句，至少有一个文字的取值为 True。我们将与该真文字对应的外部边定向为从子句顶点指向外部（进入其变量构件） v_C 。其余边均定向指向 v_C。由于进入 v_C 的边最多有两条，其总入射负载至多为 $2\cdot(1-\epsilon)<2$。

2. 如果变量 x为真，我们就将其环按顺时针方向定向。这意味着权重为 2 的内部边从 v_x^i 指向 $v_{\bar{x}}^i$，而权重为 $1+\epsilon$ 的内部边从 $v_{\bar{x}}^i$ 指向 v_x^{i+1}。此外，表示否定文字 $\bar{x}$ 的所有外部边都从环向外定向（指向各自的子句），而表示 x 的外部边则定向为指向环内。

我们计算任意环顶点 u的入射负载：(*i*) 如果 u 与一条从环向外定向的外部边相接，则有一条权重为 2 的边指向它。(*ii*) 如果 u 与一条（权重为 $1-\epsilon$ ）指向环内的外部边相接，则也有一条权重为 $1+\epsilon$的环边指向它。指向 u 的总负载为 $(1+\epsilon)+(1-\epsilon)=2$。如果变量为假，分析过程与此对称（将环按逆时针方向定向）。因此，所有环顶点中的最大负载恰好为 2。

对于证明的另一方向，假设存在一种最大负载为 2 的定向。

变量构件中的每个顶点 u都关联着一条权重为 2 的环边和一条权重为$1+\epsilon$的环边。环的方向必须保持一致，要么全部顺时针，要么全部逆时针，否则某个顶点的负载将至少达到 $2+(1+\epsilon)>2$。

对于任意子句顶点 v_C，如果三条权重为 $1-\epsilon$ 的关联边都指向该顶点，其总负载将为 $3-3\epsilon > 2$。由于最大负载为 2，每个子句顶点都必须至少有一条关联边从该顶点指出（指向某个变量构件）。

该定向会导出一个真值赋值：如果某个变量的环定向后，顶点 v_i^x 由权重为 $(1+\epsilon)$ 的边进入，那么它的正文字边也可以进入该环，我们便将其设为 True；否则，将其设为 False。由于这些环完全按顺时针或逆时针方向定向，文字赋值之间不会发生冲突，因此得到的真值赋值完全一致。由于每个子句至少有一条边从其顶点引出，因此每个子句都得到满足。

■

接下来，我们证明，能够揭示哪些顶点接收到权重严格大于 $\frac{1}{2}C^*_{max}$ 的作业的预言机，并不能实现优于 $\frac{3}{2}-\epsilon$ 的近似。首先注意，与子句顶点相邻的所有边的权重均为 $1-\epsilon<1$。由于 $C^*_{max}\geq 2$，预言机不会向任何子句顶点提供信息。

如果 $C^*_{max}=2$，那么每个变量环都必须朝同一个方向定向。由于所有环边的权重均为 >1，无论最优赋值（以及相应的真值赋值）如何，每个环顶点恰好有一条权重为 >1 的入边。因此，预言机会向所有变量顶点提供信息。由于这些信息与解无关，而且即使已知存在真值赋值，3-SAT 仍然是 NP困难的，因此预言机不会提供任何有用信息。

最后，如果 $C^*_{max}>2$，那么 $C^*_{max}\geq 3-\epsilon$，这意味着存在$\frac{3-\epsilon}{2}$ 的间隙。由于 ϵ 可以任意小，因此没有算法能够保证低于 $3\ 2$ 的近似比。

为了将证明扩展到 $\alpha<\frac{1}{2}$，令 $\epsilon<\min(1\,3,1-2\alpha)$，并考虑相同的归约，但使用一个 3SAT公式，该公式具有这样的性质：不存在满足赋值能使某个子句中的所有文字都满足。众所周知，即使将输入限制为这类实例，该问题仍然是 NP困难 [16]。注意，对于这类实例，至少有一条边必须是入边，指向子句顶点，因此在某个最优解中，所有顶点都有一条权重至少为 $1-\epsilon>\alpha C^*_{max}$ 的入边。因此，这些额外信息无法带来优于 $\frac{3}{2}-\epsilon$ 的近似。

■

接下来我们将证明，只要预言机揭示权重大于 αC^*_{max} 的边的信息，对于任意 $\frac{1}{2}<\alpha<1$，即使可以访问预言机 LM，Graph Balancing 仍然是 NP 困难的，且难以近似。

定理 6.4 即使预言机会揭示某个最优解中的 C^*_{max} 的值，以及具有权重大于 αC^*_{max} 的入边的顶点集合，对于任意 $\frac{1}{2}<\alpha<1$，在 Graph Balancing 问题中将最小完工时间近似到小于 $2-\alpha$ 倍仍是 *NP* 困难的。

证明： 我们采用与定理 6.2 的证明类似的构造。令 $\alpha'=\alpha+\epsilon$，其中 $0<\epsilon<\frac{1-\alpha}{2}$。区别如下：(i) 变量构件中的环边权重在 2 和 $2\alpha'$ 之间交替。(ii) 连接子句顶点与变量构件的所有边的权重均为 $2-2\alpha'<2\alpha$。(iii) 每个子句顶点都附加一条权重为 $4\alpha-2<2\alpha$ 的自环。按照定理 6.2 的证明，当且仅当该 3SAT 实例存在满足赋值时，图平衡实例才存在最大负载为 2 的定向。注意，变量构件中的每条环边的权重均为 2 或 $2\alpha'=2\alpha+2\epsilon>2\alpha$；对于每个子句顶点，其所有关联边的权重均为 $2-2\alpha'<2-2\alpha<2\alpha$（因为 $\alpha>\frac{1}{2}$），或者为 $4\alpha'-2=4\alpha-2+4\epsilon<4\alpha-2+2(1-\alpha)=2\alpha$。因此，在任何满足 $C^*_{max}=2$ 的解中，变量构件中的每个顶点都必须有一条权重至少为 2α 的入边。显然，没有任何子句顶点能够有一条权重至少为 2α 的入边。由此可知，预言机会告知变量构件中的所有顶点，却不会告知任何子句变量，因此预言机毫无用处。

如果 $C^*_{max}>2$，那么 $C^*_{max}\geq 4-2\alpha$，这意味着间隙为 $2-\alpha$。

■

最后，我们证明了一个关于近似 SANTA CLAUS问题（最大最小公平分配）的困难性结果，即使算法可以调用预言机$_{\mathsf{LM}}$。具体而言，我们证明，即使算法可以调用预言机$_{\mathsf{LM}}$，也无法达到优于 2-近似的结果。由于我们调用预言机$_{\mathsf{LM}}$ 的算法能够实现 2-近似（当 α 固定为 2 时），因此这个下界是紧的。我们的困难性结果针对的是一类受限的 SANTA CLAUS，对应于 GRAPH BALANCING：每个作业的长度固定，且只能分配给至多两台机器。因此，用图平衡的术语来说，我们的目标是为图的边定向，使进入某个顶点的最小权重最大化。

定理 6.5 对于任意 $\alpha \leq \frac{1}{2}$，将 MAX-MIN GRAPH BALANCING 问题近似到小于2的因子是*NP*困难的，即使有一个预言机能够揭示 C^*_{min} 的值，以及在某个最优解中具有权重大于 αC^*_{min} 的入边的顶点集合。

证明： 我们的归约来自 3-SAT。给定一个 3-SAT 公式，我们构造一个 MAX-MINGRAPH BALANCING 实例，使得当且仅当存在一个满足赋值时，存在一个最小负载为 $C^*_{min} = 2$ 的定向。此外，接收权重大于 1 的边的顶点集合与解无关。

令 $0 < \epsilon < \frac{1}{3}$。对于每个子句 $C = (\ell_1 \vee \ell_2 \vee \ell_3)$，我们创建一个**子句顶点** v_C。它有一个权重为 1 的自环，以及三条关联边，将其连接到分别对应于文字 ℓ_1、ℓ_2 和 ℓ_3 的变量构件。连接子句顶点与变量构件的所有边的权重均为 $1+\epsilon$。

对于每个变量 x，令 $k = \max(\text{count}(x), \text{count}(\bar{x}))$ 表示文字 x 或其否定 $\bar{x}$ 在公式中出现的最大次数。x 的**变量构件** 由一个长度为 $2k$ 的环组成。图 10 给出了 $k = 3$ 的一个示例。

- 环上的顶点根据其承载的文字交替排列：v^1_x，$v^1_{\bar{x}}$、v^2_x，$v^2_{\bar{x}}$，……，v^k_x，$v^k_{\bar{x}}$。
- 环边的权重在 2 和 $1-\epsilon$ 之间交替。它们的排列方式是：沿顺时针方向移动时，权重为 $1-\epsilon$ 的边总是进入一个 x-节点，而权重为 2 的边总是进入一个 $\bar{x}$-节点。
- 公式中的每次文字出现都会贡献一条权重为 $1+\epsilon$ 的边，将环上一个可用的同类型顶点与相应的子句顶点连接起来。
- 如果文字出现次数不相等（例如，$\text{count}(x) = 2$ 和 $\text{count}(\bar{x}) = 3$，即 $k = 3$），我们仍然构造一个包含 $2k$ 个节点、节点类型交替排列的完整环。$|\text{count}(x) - \text{count}(\bar{x})|$ 对应出现次数较少的文字的顶点带有一条权重为 $1+\epsilon$ 的自环。

命题 6.6 一个 3-SAT 公式可满足，当且仅当相应的圣诞老人问题实例存在一种定向，使其最小负载为 2。

证明： 假设存在一个满足公式的真值赋值。我们按如下方式对图进行定向：

1. 对于每个子句，至少有一个文字的取值为真。我们将对应于真文字的外部边定向为指向子句顶点 v_C（从其变量构件引出）。其余边则从 v_C 向外定向。加上自环边后，指向 v_C 的总入射负载至少为 $2+\epsilon > 2$。
2. 如果变量 x 为真，我们就将它的环定向为逆时针。这意味着权重为 2 的内部边从 $v^i_{\bar{x}}$ 指向 v^i_x，而权重为 $1-\epsilon$ 的内部边从 v^{i+1}_x指向 $v^i_{\bar{x}}$。此外，

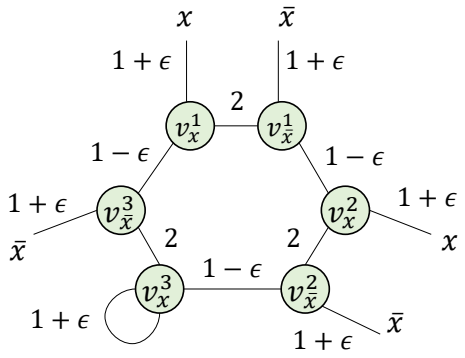

图 10：3-SAT 公式中变量 x 的变量构件，其中 x 出现 2 次，$\bar{x}$ 出现 3 次。五个输出端分别连接到对应的子句顶点。其中一个输出端是自环。

所有表示否定文字 $\bar{x}$ 的外部边都指向环内（从各自的子句指向环），而表示 x 的外部边则指向环外。

我们计算任意环顶点 u 的入射负载：(*i*) 如果 u 与一条从环向外定向的外部边相接，则它由一条权重为 2 的边进入。(*ii*) 如果 u 与一条（权重为 $1+\epsilon$）指向环内的外部边相接，则它也由一条权重为 $1-\epsilon$ 的环边进入。进入 u 的总负载为 $(1+\epsilon)+(1-\epsilon)=2$。(*iii*) 如果 u 与权重为 $1+\epsilon$ 的自环相接，则它也由一条权重为 $1-\epsilon$ 或 2 的环边进入。进入 u 的总负载至少为 2。

如果变量为 False，分析过程是对称的（将环定向为顺时针）。因此，所有环顶点中的最小负载恰好为 2。

对于证明的另一个方向，假设存在一种定向，使最小负载为 2。

考虑一个变量 x。x 的变量构件中的每个顶点都关联一条权重为 2 的环边、一条权重为 $1-\epsilon$ 的环边，以及一条权重为 $1+\epsilon$ 的外边。环必须统一定向为顺时针或逆时针；否则，某个顶点收到的负载至多为$1+\epsilon<2$。

对于任意子句顶点 v_C，自环显然指向该顶点。要使入射负载至少为 2，三条权重为 $1+\epsilon$ 的关联边中至少有一条必须指向该顶点（从变量构件指向外部）。

该定向会导出一个真值赋值：如果某个变量的环定向使得顶点 v_i^x 由权重为 $1-\epsilon$ 的边指向，那么它的正文字边也必须指向环内，我们将该变量设为 False；否则，将其设为 True。由于各个环都统一定向为顺时针或逆时针，文字赋值之间不会发生冲突，因此得到的真值赋值完全一致。由于每个子句至少有一条变量边指向它，因此每个子句都得到满足。

■

接下来，我们证明，对于 $\alpha \le \frac{1}{2}$，即使预言机揭示了哪些顶点分配到了权重严格大于αC^*_{min} 的作业，也无法实现优于$2-\epsilon$ 的近似。变量构件上的权重意味着 $C^*_{min} \le 2$。

假设 $C^*_{min}=2$。首先请注意，每个子句顶点 v_C 都与四条边相邻，其中一条边的权重为 1，另外三条边的权重为 $1+\epsilon$。由于 $C^*_{min}=2$，至少有一条权重为 >1 的边进入 v_C。由于对于任意 $\alpha \le \frac{1}{2}$，都有 $\alpha C^*_{min} \le 1$，预言机将告知所有子句顶点。

已知 $C^*_{min} = 2$，每个环顶点要么由一条权重为 2 的边进入，要么由一条权重为$1+\epsilon$的边进入。因此，无论最优赋值如何（以及相应的真值赋值如何），每个环顶点都恰好有一条权重为 $> 1 \geq \alpha C^*_{min}$ 的入边。因此，预言机将告知所有变量顶点。由于这些信息与解无关，并且即使已知存在真值赋值， 3-SAT仍是NP困难问题，预言机不会提供任何有用信息。

最后，如果 $C^*_{min} < 2$，那么 $C^*_{min} \leq 1 + \epsilon$，这意味着差距为 $\frac{2}{1+\epsilon}$。由于 ϵ 可以任意小，因此没有算法能够保证低于 2 的近似比。 ■

定理 6.7 对于任意 $\frac{1}{2} < \alpha < 1$，即使预言机能够揭示 $C^*_{\min}$ 的值以及在某个最优解中具有权重大于 $\alpha C^*_{\min}$ 的入边的顶点集合，要在因子小于 $\frac{1}{\alpha}$ 的范围内近似 Max-Min Graph Balancing 问题仍是*NP*困难。

证明： 令 $0 < \epsilon < 1 - \alpha$ 并设 $\alpha' = \alpha + \epsilon$，使得 $0 < 2 - 2\alpha' < 1 < 2\alpha < 2\alpha' < 2$。我们采用与定理 6.5 的证明中类似的构造。区别如下：(i) 变量构件中的环边权重在 2 和 $2 - 2\alpha'$ 之间交替。(ii) 连接子句顶点与变量构件的所有边，以及未匹配变量顶点上的自环，权重均为 $2\alpha'$。(iii) 每个子句顶点上的自环权重为 $2 - 2\alpha'$。

按照定理 6.5 的证明，当且仅当 3-SAT 实例存在满足赋值时， Max-Min Graph Balancing 实例存在最小负载为 $C^*_{min} = 2$ 的定向。此外，如果$C^*_{min} = 2$，那么每个子句顶点都会收到至少一条权重为 $2\alpha' > 2\alpha$的入边，而每个变量顶点都会收到一条权重为 2 或 $2\alpha'$ 的入边（两者都严格大于$\alpha C^*_{\min} = 2\alpha$）。因此，无论最优定向如何，预言机都会告知图中的所有顶点，这使得预言机毫无用处。

最后，如果 $C^*_{\min} < 2$，那么至少有一个顶点的入射负载严格小于 2，这意味着$C^*_{\min} \leq \max\{2 - 2\alpha', 2\alpha'\} = 2\alpha'$，从而得到一个 $\frac{2}{2\alpha'} = \frac{1}{\alpha+\epsilon}$的间隙。由于$\epsilon > 0$可以任意小，因此没有多项式时间算法能够保证近似比严格小于 $\frac{1}{\alpha}$。 ■

7 讨论

本文探讨了在无关并行机上为经典调度问题使用可信建议。我们的研究结果表明，即使是受到严格限制的预言机建议，也能成功突破长期存在的最坏情况下的近似障碍。我们定义了几个预言机，并证明，只需向我们最弱的预言机 $\mathsf{Oracle}_{\mathsf{LM}}$ 查询一次——它仅指出处理长作业的机器子集——就能将 Min Makespan 的近似从经典近似比 2 的障碍提升至 $\sqrt{3}$。我们进一步改进了这些技术，从而为该问题的其他变体取得了更好的近似比。通过对偶性，我们还确定了可信建议的理论极限，表明这些调度问题仍然是 NP困难的，无法获得近似；在某些情况下，甚至能达到已知的不使用预言机时的下界。

一个自然的下一步是进一步缩小我们提出的新上界与经典下界之间的差距。尽管本研究聚焦于提供长作业分配信息的预言机，但探索能够提供不同类型信息的可信建议也是一项挑战。此外，将可信建议应用于其他问题，例如带优先约束或作业截止日期的调度，也是一个很有前景的方向。沿着这一思路，我们已经证明，我们的预言机能够改进广义指派问题（GAP）的近似比。

最后，我们的算法只进行一次限制极为严格的预言机查询。探究允许的查询次数与由此获得的近似保证之间的权衡，或许能进一步揭示可信建议的作用。

参考文献

[1] Sanjeev Arora。利用建议的近似算法。载于 Klaus Jansen 和 Samir Khuller（编），组合优化近似算法：第三届国际研讨会，*APPROX 2000*，《计算机科学讲义》第1913卷，第1页。Springer，2000年。[2] Arash Asadpour、Uriel Feige 和 Amin Saberi。圣诞老人问题与超图匹配。《*ACM*算法汇刊》，8(3)，2012年7月。[3] Arash Asadpour 和 Amin Saberi。不可分割物品最大最小公平分配的近似算法。*SIAM Journal on Computing*，39(7)：2970–2989，2010年。[4] Yuichi Asahiro、Jesper Jansson、Eiji Miyano、Hirotaka Ono 和 Kouhei Zenmyo。使最大加权出度最小化的图定向近似算法。*Journal of Com-binatorial Optimization*，22(1)：78–96，2011年。[5] Kaito Baba、Evripidis Ba mpis 和 Giorgos Mitropoulos。无关机器最大完工时间调度的学习增强近似。*arXiv*预印本 *arXiv:2606.13133*，2026年。[6] N. Bansal 和 M. Sviridenko。圣诞老人问题。载于第*38*届*ACM*计算理论年度研讨会（*STOC*）论文集，第31–40页，纽约，2006年。ACM。[7] Ivona Bez´akov´a 和 Varsha Dani。不可分割物品分配。《*SIGecom*交流》，5(3)：11–18，2005年。[8] Joan Boyar、Lene M. Favrholdt、Christian Kudahl、Kim S. Larsen 和 Jesper W. Mikkelsen。带建议的在线算法：综述。《*ACM*计算调查》，50(2)：19:1–19:34，2017年。[9] Deeparnab Chakrabarty、Julia Chuzhoy 和 Sanjeev Khanna。关于如何分配物品以最大化公平性。载于第*50*届*IEEE*计算机科学基础年度研讨会（*FOCS*）论文集，第107–116页。IEEE计算机学会，2009年。[10] Tom´aˇs Ebenlendr、Marek Krˇc´al 和 Jiˇr´ı Sgall。图平衡：无关并行机调度的一个特例。《*Algorithmica*》，68(1)：62–80，2014年。[11] Yuval Emek、Pierre Fraigniaud、Amos Korman 和 Adi Ros´en。带建议的在线计算。《理论计算机科学》，412(24)：2642–2656，2011年。[12] Ellis Horowitz 和 Sartaj Sahni。非同质处理器调度的精确算法与近似算法。《*ACM*杂志》，23(2)：317–327，1976年。[13] Klaus Jansen 和 Lars Rohwedder。受限分配问题的准多项式近似。载于整数规划与组合优化，《计算机科学讲义》第10328卷，第305–316页。Springer，2017年。

[14] Kangbok Lee、Joseph Y.-T. Leung 和 Michael L. Pinedo。带限制的图平衡问题札记。*Information Processing Letters*，110(1)：24–29，2009年。[15] Jan Karel Lenstra、David B Shmoys 和 ´Eva Tardos。无关并行机调度的近似算法。*Mathematical programming*，46(1)：259–271，1990年。[16] Thomas J. Schaefer。可满足性问题的复杂度。载于第十届*ACM*计算理论年度研讨会（*STOC*）论文集，STOC '78，第216–226页，美国纽约州纽约市，1978年。计算机协会。[17] Evgeny V. Shchepin 和 Nodari Vakhania。最优舍入可为无关机器调度带来更好的近似。*Operations Research Letters*，33(2)：127–133，2005年。[18] Ola Svensson。圣诞老人问题在无关机器上调度作业。*SIAM Journal on Computing*，41(5)：1318–1341，2012年。[19] Michalis Xefteris。从估计到调度：学习增强的受限分配。*arXiv*预印本 *arXiv:2606.08377*，2026年。