LOTUS：一种用于增强大规模两阶段随机规划中对偶分解的热启动框架

Emma Cornielje[a,b], Berend Markhorst[a,b,*], Alessandro Zocca[b], Rob van der Mei[a,b]

[a]随机学组，*CWI*，阿姆斯特丹，荷兰[b]数学系，阿姆斯特丹自由大学，阿姆斯特丹，荷兰

arXiv:2603.01733v1 [math.OC] 2 Mar 2026

摘要

求解大规模两阶段随机混合整数规划在计算上具有挑战性。我们提出LOTUS，一种基于子集的热启动框架，可在固定时间预算下增强对偶分解。通过使用有信息的乘子初始化对偶搜索，LOTUS加速原始收敛，并部分缓解弱线性规划松弛的影响。通过对生产计划实例的广泛计算研究，我们表明，在两小时内，LOTUS在45.83%的情形中得到显著更优的原始解，而仅在4.17%的情形中被对偶分解超越。

关键词：随机规划，热启动，对偶分解，情景缩减

1. 引言

随机规划（SP）已成为在不确定性条件下进行决策的标准框架，应用领域涵盖能源和供应链管理[29]。然而，要准确表示不确定性，往往需要构建大型情景树，这使得大规模两阶段随机混合整数规划（SMIPs）因维数灾难和非凸性而在计算上难以处理 [28]。Benders分解虽然能有效处理连续补救，但随机混合整数规划通常需要采用对偶分解（DD） [19]或渐进对冲 [7]等情景分解方法。

尽管这些方法很有用，但它们在大规模问题上难以收敛。近期文献指出，热启动过程可以通过从缩减后的问题子集中生成初始解，缓解这些计算难题 [22]。

在本研究中，我们提出了 LOTUS（利用子集进行两阶段随机规划的拉格朗日优化）。LOTUS 将热启动策略融入对偶分解框架，以便在计算预算紧张的情况下找到高质量的原始解。具体而言，我们在代表性情景子集上采用次梯度下降 [4] ，生成有依据的对偶乘子，为全规模分解提供有效的起点。为严格评估 LOTUS 并推动未来研究，我们推出了一组新的大规模生产计划实例，旨在考验最先进水平的随机混合整数规划求解器。

*通讯作者电子邮箱： e.j.cornielje@student.vu.nl (Emma Cornielje), berend.markhorst@cwi.nl (Berend Markhorst), a.zocca@vu.nl (AlessandroZocca), mei@cwi.nl (Rob van der Mei)

本文其余部分安排如下：第2节回顾相关研究，第3节详细介绍 LOTUS 框架，第4节给出计算结果，表明在实际可行的时间限制内，该框架取得的原始目标值优于标准 DD。

2. 文献

两阶段随机混合整数规划因补救函数失去凸性以及场景集带来的维数灾难而 notoriously 难以求解。虽然确定性等价规划（DEP）可直接求解小规模实例，但大规模应用需要分解技术。这些技术大致分为逐阶段分解和逐情景分解。

2.1. 分解框架

逐阶段分解，尤其是L形方法 [3, 26], ，在连续补救情形下表现良好。然而，当第二阶段出现整数变量时，补救函数变为非凸且不连续。整数L形方法 [20] 通过专用切平面（例如最优性切平面）来处理这一问题，但由于这些切平面较弱以及求解混合整数子问题的计算成本，往往收敛缓慢[21]。

逐情景分解放松非预见性约束（NACs），从而将问题拆分为相互独立的情景子问题。这种结构非常适合并行计算。此类方法中有两种突出的方法：渐进对冲（PH）和对偶分解（DD）。PH [27] 采用增广拉格朗日函数

方法，并结合二次惩罚。尽管PH是处理随机混合整数规划（SMIPs）的一种强大启发式方法，但对于整数补救，它失去了收敛保证，也不会自然地产生有效的对偶界 [12]。

相反，DD [5] 通过拉格朗日松弛放松非预见性约束，而不使用二次惩罚。由此得到的拉格朗日对偶问题为最小化问题提供了有效的下界，通常通过次梯度法 [4] 或束方法 [31]求解。然而，对于SMIPs而言，强对偶性并不一定成立；拉格朗日对偶的最优值通常为原始最优值提供下界，并且由于整数可行域的非凸性，往往会产生对偶间隙。为弥合这一差距并恢复原始解，DD通常嵌入分支定界框架（拉格朗日分支定界）[1]中。尽管标准DD在理论上是可靠的，但对偶乘子的缓慢收敛常常阻碍其应用。

2.2. 加速收敛：热启动与归约

为应对精确方法带来的计算负担，近期文献将重点放在热启动策略 [8, 22]上。其思想是通过松弛问题或考虑情景子集，使用问题的简化版本为全规模问题生成先进的初始信息。

已有多种框架成功采用了这一思路。对于内点法，[8] 证明，由归约后的情景树得到的解可作为高质量的初始点。在分支割（B&C）中，TULIP框架 [22] 采用两步法：先求解由部分情景诱导的问题，以生成有效不等式，再以此热启动完整问题。LOTUS采用多步框架，但适用范围更广，可用于更多类型的两阶段优化问题。此外，[14]提出用渐进对冲法热启动对偶分解（DD），利用渐进对冲法快速推进启发式求解的优势，为精确的DD算法初始化拉格朗日乘子。

我们提出的框架LOTUS利用情景缩减来生成热启动。情景缩减旨在用支撑更小的离散测度近似原始概率测度。通过用代表性子集近似完整的不确定集，LOTUS将搜索聚焦于问题的核心权衡。这滤除了通常导致主问题振荡的次要情景变化，使算法能够绕过缓慢的初始搜索，并立即识别出高质量的第一阶段决策，以实现高质量的原始值恢复。情景缩减方法分为基于分布 或基于问题两类。基于分布的方法，例如快速前向选择 [16], 概率距离最小化 [10], 或矩匹配 [2], ，选择情景以最小化统计距离（例如Wasserstein度量），而不考虑当前问题的优化结构。尽管计算速度很快，它们可能会舍弃对目标函数具有重大影响的低概率情景。

相比之下，问题驱动方法（综述见 [6]）在归约时会考虑底层优化问题的结构。这一思路已应用于多种算法，例如 [17, 11, 23]。LOTUS采用一种兼具基于分布和问题驱动特性的归约技术来热启动对偶分解过程，从而在实际时间限制内更容易获得高质量的原始解。

3. 方法论

本节介绍 LOTUS 框架。首先，在第 3.1 小节中，我们将逐步说明 LOTUS 算法。在第 3.2 小节中，我们将介绍 LOTUS 框架的假设与设计动机。

在说明我们的方法论之前，我们先简要介绍两阶段随机混合整数规划（DEP）的确定性等价问题表述。

$$
\begin{aligned}
\min_{x,\{y_{(s)}\}_{s\in\mathcal{S}}} \quad & c^T x + \sum_{s\in\mathcal{S}} p_{(s)} q_{(s)}^T y_{(s)} && && \text{(1a)}\\
\text{s.t.} \quad & Ax \geq b, && && \text{(1b)}\\
& T_{(s)}x + W_{(s)}y_{(s)} \geq h_{(s)}, && \forall s \in \mathcal{S}, && \text{(1c)}\\
& x \in \mathbb{Z}_+^{n_1} \times \mathbb{R}_+^{n_2}, && && \text{(1d)}\\
& y_{(s)} \in \mathbb{Z}_+^{k_1} \times \mathbb{R}_+^{k_2}, && \forall s \in \mathcal{S}. && \text{(1e)}
\end{aligned}
$$

这里，$\mathcal{S}$ 表示情景集合，其中每个情景 $s \in \mathcal{S}$ 以概率 $p_{(s)}$ 出现，且 $\sum_{s\in\mathcal{S}} p_{(s)} = 1$。第一阶段决策变量向量为 x，其成本向量为c。第一阶段约束由矩阵 A和右端项 b定义。对于每个情景 s，$y_{(s)}$ 表示第二阶段补救变量，其成本向量为 $q_{(s)}$。第二阶段约束由补救矩阵$W_{(s)}$、技术矩阵 $T_{(s)}$ 和右端项$h_{(s)}$定义。请注意，这三者不一定是随机的。(1d) 和 (1e) 中定义的决策域表示混合整数要求，其中变量总数分别为 $n = n_1 + n_2$ 和 $k = k_1 + k_2$。

3.1. 方法

图1展示了 LOTUS 框架的示意概览。在该概览中，红色方框对应原问题，橙色方框对应完整问题的拉格朗日松弛形式，绿色方框对应约化问题的拉格朗日松弛形式。此外，虚线箭头表示框架中采取的步骤。以下各节将更详细地讨论 LOTUS 中应用的每个步骤。

3.1.1. 步骤 *1*：对偶分解

我们将对偶分解应用于两阶段随机规划模型。为了解耦各情景，我们引入局部副本，即

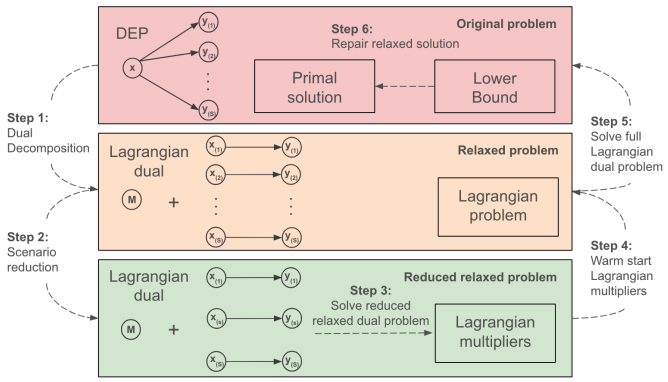

图1：LOTUS 的示意图。虚线箭头表示该方法中的步骤。红色、橙色和绿色方框分别表示原始问题、完全松弛问题和约化松弛问题。

每个情景 $s \in \mathcal{S}$的第一阶段变量$x_{(s)}$。我们强制执行显式非预见性约束：

$$x_{(s)} = x, \quad \forall s \in \mathcal{S}.$$

通过使用拉格朗日乘子 $\lambda_{(s)}$ 松弛这些非预见性约束，我们得到拉格朗日函数：

$$L(x, x_{(s)}, y_{(s)}, \lambda) = c^T x + \sum_{s \in \mathcal{S}} \left(p_{(s)} q_{(s)}^T y_{(s)} + \lambda_{(s)}^T (x_{(s)} - x) \right).$$

重新整理各项，并对 x、 $x_{(s)}$ 和 $y_{(s)}$ 最小化拉格朗日函数后，我们得到的拉格朗日函数可分解为关于全局变量 x 的主问题 (M)，以及 $|\mathcal{S}|$ 个相互独立的情景子问题 （关于 $(x_{(s)}, y_{(s)})$ ：

$$(\mathrm{M}): \quad \min_{x \in \mathcal{X}} \left(c - \sum_{s \in \mathcal{S}} \lambda_{(s)} \right)^T x, \tag{2a}$$

$$(\mathrm{S})_s: \quad \min_{x_{(s)}, y_{(s)}} \left(p_{(s)} q_{(s)}^T y_{(s)} + \lambda_{(s)}^T x_{(s)} \right). \tag{2b}$$

主问题 (2a) 在紧致集 $\mathcal{X}$ 上最小化净第一阶段成本（该集合表示第一阶段可行域，例如 $Ax \geq b$），从而避免无界性。对于每个 $s \in \mathcal{S}$，子问题 (2b) 均在各自的情景约束下求解。

为使后续算法步骤更加紧凑，我们将任意情景子集 $\mathcal{S}' \subset \mathcal{S}$ 对应的拉格朗日函数 $Z_D^{\mathcal{S}'}(\lambda)$ 定义为将拉格朗日函数限制在该子集上后得到的最小值：

$$Z_D^{\mathcal{S}'}(\lambda) = \min_{x \in \mathcal{X}, \{x_{(s)}, y_{(s)}\}_{s \in \mathcal{S}'}} L_{\mathcal{S}'}(x, x_{(s)}, y_{(s)}, \lambda). \tag{3}$$

因此，完整的拉格朗日对偶问题通过求解以下问题来寻求最紧的下界：

$$\max_{\lambda \in \mathbb{R}^{|\mathcal{S}|}} Z_D^{\mathcal{S}}(\lambda).$$

3.1.2. 步骤 *2*：情景缩减（热启动）

情景缩减是LOTUS框架的核心创新。对于大规模情景集合 $\mathcal{S}$ ，求解完整的拉格朗日对偶问题在计算上难以承受。为加速收敛，LOTUS通过在规模小得多的代表性情景子集 $\mathcal{S}' \subset \mathcal{S}$上求解对偶问题，为拉格朗日乘子向量 λ 生成热启动 ，其中 $|\mathcal{S}'| \ll |\mathcal{S}|$。

该热启动的质量取决于 $\mathcal{S}'$ 对完整不确定集的近似准确程度。我们研究结合两种度量来选择 $\mathcal{S}'$：

基于分布的度量。 令 ξ_s 表示情景 s的随机参数向量。这种标准方法仅通过每个情景的数据向量ξ_s来刻画情景。情景之间的距离定义为欧几里得距离 $\|\xi_s - \xi_{s'}\|_2$。虽然这种度量在计算上高效，但它假设数据空间 中的邻近意味着解空间中的邻近；对于混合整数问题而言，这一假设可能不成立，因为数据的微小扰动就可能显著改变最优基。

基于问题的度量。 为解决纯粹数据驱动度量的局限性，我们引入一种基于问题的度量，以捕捉问题的成本结构。在混合整数规划问题中，数据的微小变化可能导致目标值出现较大的不连续变化（例如，超过容量阈值后产生固定成本）。因此，我们将每个情景 s 映射为一个标量特征 v_s^{LP}，表示其线性规划松弛的最优目标值。该值通过分别求解情景子问题获得（忽略非预见性约束），并放松所有整数约束。随后，情景之间（例如 s 和 s'）的距离根据这些结构代理值之间的差异 $|v_s^{LP} - v_{s'}^{LP}|$来定义。

结合度量。我们将数据向量 ξ_s 和特征 v_s^{LP}合并为一个特征向量 $\mathcal{F}_s$。为确保基于分布的特征与基于问题的特征在统计上保持平衡，我们在拼接前使用 Z-分数归一化对二者进行标准化。为确保基于问题的度量与基于分布的度量在选择算法中具有相同权重，我们为归一化后的成本特征赋予权重 ω ，其值等于家具类型的数量（$|F|$）。这一平衡处理确保选择算法在多维需求数据的条件下，将情景的潜在成本影响纳入考虑，并赋予其相同的总优先级。

选择算法。 随后，我们采用快速前向选择算法，通过迭代选择使原始分布与缩减分布之间的概率距离（例如康托罗维奇距离）最小的情景，逐步构建子集 $\mathcal{S}'$。确定 $\mathcal{S}'$ 后，重新划分原始集合 $\mathcal{S}$： 将每个被舍弃的情景 $k \in \mathcal{S} \setminus \mathcal{S}'$映射到与其欧几里得距离度量 $\|\mathcal{F}_s - \mathcal{F}_{s'}\|_2$下最近的代表性情景 $s \in \mathcal{S}'$ ，并汇总其概率，以保留总概率质量。

3.1.3. 步骤 *3*：求解约化拉格朗日对偶

在此步骤中，我们求解仅限于代表性情景子集 $\mathcal{S}' \subset \mathcal{S}$的拉格朗日对偶问题：

$$\max_{\lambda \in \mathbb{R}^{|\mathcal{S}|}} Z_D^{\mathcal{S}'}(\lambda).$$

使用步骤 1 中定义的符号，这一步旨在为约化问题寻找最紧的下界。由于 $Z_D^{\mathcal{S}'}(\lambda)$ 是凹且不可微的，我们采用 次梯度上升算法 [4]。详情请参阅附录 A。

3.1.4. 步骤 *4*：热启动初始化

将步骤 3 中从约化问题得到的优化乘子 λ^* 映射到全场景集 $\mathcal{S}$，以初始化全规模对偶问题。设 $\rho : \mathcal{S} \to \mathcal{S}'$为情景缩减步骤中的映射函数，其中，对于任意原始情景 $i \in \mathcal{S}$，$\rho(i)$ 表示 $\mathcal{S}'$ 中的代表性情景。完整问题的初始乘子 λ^0 设为：对于每个 $i \in \mathcal{S}$，均有 $\lambda^0_{(i)} = \lambda^*_{(\rho(i))}$。此热启动配置为应用于完整拉格朗日对偶问题的次梯度上升算法提供初始点。

3.1.5. 步骤*5*：求解完整拉格朗日对偶

现在，我们在完整场景集 $\mathcal{S}$上求解全规模拉格朗日对偶问题：

$$\max_{\lambda} Z_D^{\mathcal{S}}(\lambda).$$

初始化（热启动）。与步骤3不同，我们使用从约化问题中获得的信息。我们使用步骤4中构造的热启动向量 λ^{WS} 初始化乘子，即 $\lambda^0 = \lambda^{\mathrm{WS}}$。与零向量相比，这一对偶空间中的高级起始点预计会更接近最优乘子，从而能够在优化过程中更早找到更优的可行解，改善原始收敛。

算法。其余算法流程（次梯度上升、Polyak 步长和终止准则）与步骤3中所述相同，从而确保以一致的方法求解完整对偶问题。令 λ^* 表示算法终止时返回的最优（或当前找到的最佳）乘子。

3.1.6. 步骤*6*：原始恢复与评估

虽然拉格朗日对偶函数 $Z_D^{\mathcal{S}}(\lambda^*)$ 为最优值提供了严格的下界，但由此得到的特定情景解 $\{x^*_{(s)}(\lambda^*)\}_{s\in\mathcal{S}}$ 通常会违反非预见性（即 $x^*_{(s)} \neq x^*_{(s')}$）。为恢复可行的原始值解 $Z_P^{\mathcal{S}}(\lambda^*)$（从而为最小化问题提供有效的上界），我们利用主问题中的全局决策$\tilde{x}$，对完整 DEP 进行受限的原始搜索。

在此步骤中，我们求解 DEP，并将第一阶段变量限制在一个狭窄的可行性窗口（$1\pm\epsilon$）$\tilde{x}$ 其中 $\epsilon = 0.05$。此方法利用对偶搜索的进展来剪枝原始空间，使求解器聚焦于第一阶段定义域中最有希望的区域 $\mathcal{X}$。根据相对完全补救的假设（参见第3.2节），可以确保该受限 DEP 对每个情景都能找到最优的第二阶段补救变量 $\{y_{(s)}\}_{s\in\mathcal{S}}$。由此得到的目标值提供了一个有效的原始界 $Z_P^{\mathcal{S}}$，我们用它来评估最终的最优性间隙。

关于最优性的说明。如果原始界与对偶界之间的差距非零，则无法保证解是最优的。为确保非凸随机混合整数规划的精确收敛，LOTUS 被设计为一种界定过程，可嵌入例如全局分支定界方案 [1] 中。LOTUS 所带来的原始收敛加速，有助于在搜索树早期进行更积极的剪枝，从而减少探索的节点数，并缓解对偶界收敛缓慢所造成的计算负担。

3.2. 理论假设与实际动机

LOTUS框架的设计源于两阶段随机混合整数规划的特定结构性质。本节概述 LOTUS 的理论条件，以及促使我们提出热启动方法的实际挑战。

3.2.1. 理论假设

对于 LOTUS，我们对问题结构作出以下假设。

块角结构。 我们假设约束矩阵具有块角结构，其中各情景仅通过第一阶段决策变量 x相互关联。这种可分性是对非预见性约束进行拉格朗日松弛、将 DEP 分解为相互独立且易于处理的子问题的根本性质。

相对完全补救与紧致性。 我们假设问题具有相对完全补救，且可行域 $\mathcal{X}$ 和 $\mathcal{Y}_s$ 是紧致的（闭且有界）。这些性质至关重要，原因有二：

1. 算法稳定性：对于所有 $\lambda \in \mathbb{R}^n$，拉格朗日对偶函数 $Z_D^{\mathcal{S}}(\lambda)$都有良好定义 且为有限值。这可以防止目标函数中的无限惩罚或无界解导致次梯度更新不稳定。2. 原始恢复：这保证了步骤6中的投影启发式能够得到一个可行解，因为将x 固定为可行的 $\hat{x}$ 不会导致第二阶段不可行。

3.2.2. 实际动机

除了结构有效性之外，LOTUS框架还专门用于解决标准求解器和经典分解方法在求解随机混合整数规划时的局限性。

缩小整性间隙。 标准分支定界求解器依赖线性规划松弛来剪枝搜索树。然而，随机混合整数规划往往存在弱线性规划松弛，此时松弛线性规划给出的对偶界（下界） $Z_D^{\mathcal{S}}$ 与最优整数值 $Z_I^{\mathcal{S}}$相差甚远。这种较大的整性间隙会导致剪枝效果不佳，运行时间难以承受。对偶分解的动机在于，拉格朗日对偶界 $Z_D^{\mathcal{S}}$ 严格优于（或至少等于）线性规划松弛界 $Z_{LP}^{\mathcal{S}}$:

$$Z_{LP}^{\mathcal{S}} \leq Z_D^{\mathcal{S}} \leq Z_I^{\mathcal{S}} \leq Z_P^{\mathcal{S}}.$$

通过求解保留整数约束的子问题，LOTUS 得到了高质量的对偶界 $Z_D^{\mathcal{S}}$。更重要的是，该框架利用稳定的乘子指导，优先寻找高质量的可行解（$Z_P^{\mathcal{S}}$），即使理论对偶界仍相对宽松，也能从上方有效缩小最优性差距。

通过热启动加速原始收敛。标准次梯度法的收敛速度出了名地慢，主要是因为存在“识别阶段”：它们通常会耗费许多初始迭代，仅仅用于识别最优乘子向量 λ^* 中的非零分量（即活动流形）。LOTUS 通过在计算成本更低的约化问题 $Z_D^{\mathcal{S}'}$ （λ）上执行这一识别过程，绕过了这一瓶颈。由此得到的热启动λ^{WS} 有效地使全规模算法跳过初始搜索阶段，从更接近最优活动集的邻域处开始运行。这种接近性减少了冷启动次梯度法中常见的剧烈振荡，为原始启发式提供了更稳定的指导。

4. 结果

4.1. 实验设置

为了实证评估 LOTUS 算法相对于对偶分解的性能，我们在生产/制造问题的背景下开展了一项两阶段计算案例研究（详见附录 B）。

4.2. LOTUS 的实证表现

我们将 LOTUS 框架与标准DD基线进行性能比较。主要目标是评估热启动策略能否在实际的 120 分钟时限内显著提升找到高质量原始解和缩小最优性间隙的效果。对于描述性指标，我们对 24 种独特配置进行稳定化分析，每种配置的结果取 5 个独立随机种子的均值。对于统计分析，我们对 $N = 111$ 原始实例 [30] 进行 Wilcoxon 符号秩检验（依据差值的秩），以确定 DD 与 LOTUS 在原始界和最优性间隙方面的中位数差异是否显著异于零。请注意，有 9 个实例未纳入统计分析。

这些实例中，DD 未能找到可行的正原始解。

下表1显示，LOTUS 实现了 0.93% 的平均利润提升。95% 置信区间证实了这一改进的稳健性，因为整个区间严格大于零。尽管平均利润提升看似有限，全局 p 值 < 0.001 证实 LOTUS 能够持续找到更优的原始解。LOTUS 表现出可靠性，在 95.8% 的测试配置中提供了与标准DD持平（平局）或更优（获胜）的解（获胜率 $+$ 平局率）。

表1：总体表现汇总（$N = 111$）

平均利润改进 (%)	Win 比率 (%)	Draw 比率 (%)	95%置信区间	原始值 p-值	Gap p-值
0.93	45.83	50.00	$[0.38, 1.49]$	$< \mathbf{0.001}$	0.349

然而，我们观察到目标值与最优性间隙之间存在统计差异。最优性间隙的 p-值（0.349）表明，LOTUS 不一定比对偶分解更有效地缩小间隙。这与我们的实际动机一致：LOTUS 优先通过改善原始界，自上而下地缩小间隙，即使理论上的对偶界仍相对停滞。

表2：按场景数（$|\mathcal{S}|$）划分的性能，运行时间分别为 2 小时和 4 小时。

$\lvert\mathcal{S}\rvert$	平均利润改进	Win Rate	Draw Rate	N	原始值 p-值	Gap p-值	时间
250	1.94%	83.33%	0.0%	28	**0.0015**	0.0026	2h
500	0.93%	66.67%	33.33%	29	**0.0006**	0.2085	2h
1000	0.45%	33.33%	66.67%	28	**0.0213**	0.6772	2h
2000	0.41%	16.67%	83.33%	26	0.0899	0.9898	2h
2000	0.67%	33.33%	66.67%	26	**0.0090**	0.3383	4h

表2详细展示了LOTUS在不同场景数（$|\mathcal{S}|$）下的性能。LOTUS在较小规模（$|\mathcal{S}| = 250$）时优势最为明显，利润提升1.94%，且在近83%的情况下获胜。随着问题规模增至 $|\mathcal{S}| = 2000$，在2小时运行时限内，平局的抽样频率升至83.33%，平均利润提升不再显著。将 $|\mathcal{S}| = 2000$ 实例的运行时间延长至4小时，并采用更全面的热启动阶段（30次迭代而非10次迭代），展现了该框架潜在的效能，使 p值显著达到0.009，获胜率升至33.33%。这表明，尽管较大实例需要更长的热启动阶段，但LOTUS热启动提供的卓越协调最终有助于实现更好的原始收敛。

一个关键发现是，在所有测试的情景数量下，LOTUS 的总迭代次数都高于对偶分解。在 $|\mathcal{S}| = 2000$ 下，LOTUS 总共完成了 50 次

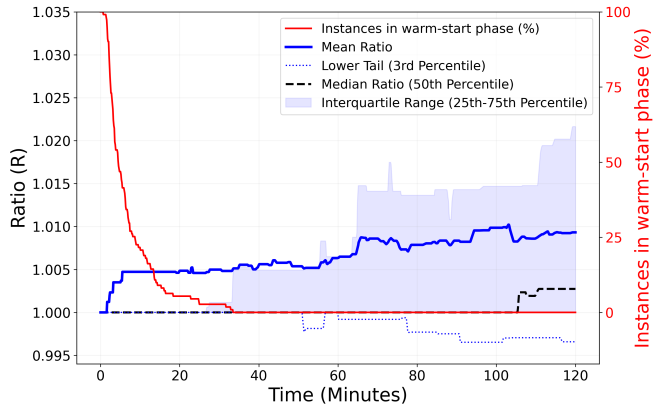

图 2：随时间变化的聚合原始界比率（$R = Z_{P,\ LOTUS}/Z_{P,\ DD}$）（蓝色）以及热启动阶段中的实例百分比（红色）。

迭代（预热阶段 10 次，主阶段 40 次），而对偶分解仅进行了 36 次迭代。多出的迭代次数证实，LOTUS 并未将时间浪费在预热上。LOTUS 似乎是以成本较低的缩减规模迭代，替代成本较高的全规模迭代。

为了解 LOTUS 在 120分钟时域内相对于对偶分解的表现，我们分析了整个运行时间内相对原始解（下界）比率的变化。图 2 展示了比率$R := Z_{P,\ LOTUS}/Z_{P,\ DD}$ 随时间的演变。图 2 显示，均值比率 R （蓝色实线）起始时恰好为 1。000，此时 100% 的实例处于热启动阶段（见红色实线）。随着实例退出热启动阶段（如红线下降所示），它们进入全规模优化阶段，此时均值比率呈上升趋势。每个实例完成各自的热启动探索后，便以更优的初始原始界开始主阶段。红线降至 0% 时，所有配置都已转为求解全场景集；此时，就均值比率 R而言，LOTUS 相比对偶分解已取得 0。48% 的平均领先优势。这证实，热启动乘子 λ^{WS} 使 LOTUS 能够克服早期的收敛平台，而标准DD会停滞于此。虽然抽样频率（45。83%）使中位数在大部分时域内保持接近 1。000，但第75百分位数在 30 分钟后显著分化。此外，2小时运行时间结束前中位数较高，表明 LOTUS 在最终收敛阶段找到了更优的解。然而，由于框架仅在 4。17% 的情况下利润较低，第25百分位数仍保持在 1。000 或以上。这意味着，LOTUS 相对于对偶分解的表现不佳仅在极端低端尾部（第3百分位数）才显现。

图2证实，对于绝大多数实例，LOTUS 至少与标准DD同样稳健，并且在近半数情况下带来了显著改进。结果还证实，LOTUS 利用了双重优势：热启动的次梯度上升迭代成本远低于完整问题中的迭代（$t_L(S') \ll t_L(S)$），而在主阶段，次梯度上升迭代的速度更快。这种综合效应带来了图2中所示的相较于DD的原始界改进，使LOTUS在同样的120分钟时限内取得了更好的原始均值和中位数。

附录C进一步分析了 LOTUS 相较于对偶分解的速度增益，并提供了更多实证分析。

5. 讨论

尽管 LOTUS 的结果令人鼓舞，但仍需承认若干局限，以便正确理解研究发现并指导后续发展。

5.1. 实验设计

一项主要局限在于，热启动配置（例如子集大小和热启动迭代次数）是根据初步实验选定的，并且统一应用于所有实例。这一做法假设配置在不同规模下具有一定的独立性，但对于极端规模的实例，这一假设可能并不成立。随着问题维度增大，热启动阶段与主要分解阶段之间的最佳平衡也会发生变化。对于规模最大的实例（$|\mathcal{S}| = 2000$），两小时的时限不足以完全弥补热启动带来的初始计算开销，导致平局频繁出现。然而，四小时时出现了显著的 p-值（0.009），这证实 LOTUS 具有提升性能的潜力，但效果取决于实例规模和分配给热启动的时间。存在这样一个“盈亏平衡点”——即通过更优的原始收敛收回低成本探索所投入时间的时刻——这表明未来有必要研究动态参数调优或自适应子集缩放。

5.2. 算法框架与基准测试

本分析在热启动阶段与主阶段均采用标准次梯度下降实现。尽管这有助于分离LOTUS的影响，但次梯度方法对步长规则较为敏感，且可能在收敛末期进展缓慢。更复杂的对偶更新机制，例如束方法[31]，，有望提供更好的稳定性并加快对偶界推进。此外，尽管标准DD是这项基础研究的合适基准测试，未来工作应将LOTUS与商业级启发式算法相结合，以便将其性能直接与工业求解器中的最先进水平进行比较。

5.3. 可推广性与问题结构

这些发现的普适性与测试实例集合的结构特性有关。生产计划实例包含常见的混合整数随机规划挑战，例如固定费用成本、第二

阶段的整数性，以及弱线性规划松弛。另一方面，定量增益（平均利润提升。93%）会受到模型特定补救结构的影响。在第一阶段决策与第二阶段决策耦合较松，或不确定性成本较低的问题中，高质量对偶热启动的影响可能不那么显著。

有必要在其他具有与第3.2节所述类似数学问题结构的领域中测试 LOTUS，以确立更普遍的性能表现。未来实验的有希望候选领域包括电力系统中的随机机组组合，其特点是显著的固定费用启动成本导致线性规划松弛较弱，并依赖基于场景的分解 [5, 24]。类似地，带容量限制的固定费用网络设计问题具有较大的整性间隙和密集耦合，而 LOTUS 正是为缓解这些问题而设计的 [13]。在这两个领域中，LOTUS 在缩减的情景集上识别稳定活动流形的能力，或许能够克服标准DD目前面临的计算瓶颈。

6. 结论

求解大规模、真实世界的两阶段随机混合整数规划，面临着诸多计算挑战，这些挑战源于其非凸性、高维性以及大规模线性规划松弛。在本研究中，我们提出并评估了面向两阶段随机规划的子集拉格朗日优化（LOTUS）框架，这是一种将基于子集的热启动策略融入对偶分解（DD）方法的新颖方案。LOTUS 通过使用从代表性情景子集中得出的有信息的乘子来初始化对偶搜索，显著提升了在实际计算时间窗口内找到高质量原始解的能力。这一效果有两个原因：首先，其热启动迭代成本低；其次，在返回求解完整问题时，它能够达到更高的迭代速率。

一项覆盖120个测试实例的广泛计算研究结果表明，LOTUS是一种稳健且实用的工业决策工具。尽管数学最优性间隙仍与标准对偶分解相当，LOTUS却能识别出显著更优的原始界（$p < 0.001$），平均利润提升达0.93%。此外，该框架的成功率为95.8%，提供了对偶分解在求解早期所缺乏的可靠性水平，而在这一阶段，稳定的引导对于整数恢复至关重要。

我们的分析揭示了性能随规模变化的特征。对于中等规模的问题，LOTUS 能立即带来收益；而对于规模最大的实例，则需要更长的运行时间，才能克服热启动阶段最初的计算开销。从两小时节点时频繁打平，到四小时节点时规模最大实例的获胜率显著提高，这一变化证实了热启动能够实现更优的协调，最终使求解器突破原始收敛平台，而对偶分解则停滞于此。

第5节中指出的局限性引出了若干有前景的未来研究方向。首先，可将LOTUS集成到完整的求解方法论中，例如分支定界或分支切割框架。LOTUS所提供的加速原始收敛有助于更早地对搜索树进行更积极的剪枝，从而减少所探索的节点数量，并减轻对偶界收敛较弱所带来的计算负担。此外，用更稳定的束方法或水平方法取代次梯度上升，有可能加速热启动阶段。

最后，开发一种自适应 LOTUS 框架，根据收敛行为动态确定热启动配置和子集大小，有望进一步优化初始热启动阶段与主执行阶段之间的时间盈亏平衡点。

致谢 本出版物部分由荷兰研究委员会（NWO）通过“Readiness”项目（*TWM.BL.019.002*）以及 A. Zocca 的个人 VIDI 资助（*VI.VIDI.233.247*）提供经费支持。作者声明不存在与本文内容相关的竞争利益。

生成式人工智能使用声明作者仅使用生成式人工智能工具来提升稿件的清晰度和语言质量。未使用这些工具生成科学内容、结果、证明、数据或解释。所有实质性内容均为作者原创，作者对稿件的准确性和完整性承担全部责任。

参考文献

[1] P. Aldenvik 和 M. S. Schierscher. 从混合二进制线性规划的对偶次梯度法中恢复原始解：一种分支定界法。2015 . doi：https://api.semanticscholar.org/CorpusID:54580758。

[2] C. Beltran-Royo。通过条件情景快速缩减两阶段随机混合整数线性规划问题中的情景。《优化方法与软件》，37(1):23–44, 2022。doi：https://doi.org/10.1080/10556788.2019.1697696。

[3] J. Benders。用于求解混合变量规划问题的分区过程。《数值数学》，4:238–252，1962年12月。doi：https://doi.org/10.1007/BF01386316。

[4] M. Bragin。混合整数线性规划的拉格朗日松弛综述：重要性、挑战、历史回顾、最新进展与机遇。*Annals of Operations Research*，333:29–45，2024。doi：https://doi.org/10.1007/s10479-023- 05499-9。

[5] C. Carøe 和 R. Schultz。随机整数规划中的对偶分解。《运筹学快报》，24(1):37–45，1999。ISSN 0167-6377。doi：https://doi.org/10.1016/S0167-6377(98)00050-9。

[6] X. Chou 和 E. Messina。面向问题的随机规划问题情景生成：综述。*Algorithms*，16(10):479，2023。

[7] J. Christiansen、B. Dandurand、A. Eberhard 和 F. Oliveira。关于随机整数规划中渐进对冲的研究。*Computational Optimization and Applications*，86：989–1034，2023 年 10 月。doi：https://doi.org/10.1007/s10589-023-00532-w。

[8] M. Colombo、J. Gondzio 和 A. Grothey。面向大规模随机线性规划的热启动方法。*Mathematical Programming*，127：371–397，2009。doi：https://doi.org/10.1007/s10107-009-0290-9。[9] E. Cornielje。Lotus-framework，2026。网址：https://github.com/ECornielje/LOTUS-framework。GitHub 仓库。[10] J. Dupačová、N. Groewe-Kuska 和 W. Roemisch。随机规划中的情景缩减：一种采用概率度量的方法。*Mathematical Programming*，95：493–，2003 年 1 月。doi：https://doi.org/10.1007/s10107-002- 0331-0。[11] Y. Feng 和 S. Ryan。用于随机机组组合的基于解灵敏度的情景削减。*Computational Management Science*，13(1)：29–62，2016 年 1 月。doi：https://doi.org/10.1007/s10287-014-0220-z。[12] D. Gade、G. Hackebeil、S. Ryan、J.-P. Watson、R. J.-B. Wets 和 D. L. Woodruff。利用渐进对冲算法获得随机混合整数规划的下界。*Mathematical Programming*，157 67，2016 年 4 月 47 10 1007 10107 016 1000：–。doi：https://doi.org/。/s - - - z。[13] B. Gendron、O. Gunluk 和 A. Lodi。用于多商品容量约束固定费用网络设计的基于节点的拉格朗日松弛。*Discrete Applied Mathematics*，2018。doi：https://doi.org/10.1016/j.dam.2020.12.024。[14] G. Guo、G. Hackebeil、S. M. Ryan、J.-P. Watson 和 D. L. Woodruff。在随机整数规划中整合渐进对冲与对偶分解。*Operations Research Letters*，43(3)：311–316，2015。ISSN 0167-6377。doi：https://doi.org/10.1016/j.orl.2015.03.008。[15] Gurobi Optimization, LLC。*Gurobi优化器参考手册*，2022。网址：https://www.gurobi.com。版本 10.0。[16] H. Heitsch 和 W. Roemisch。随机规划中的情景缩减算法。*Computational Optimization and Applications*，24：187–206，2003 年 1 月。doi：https://doi.org/10.1023/A:1021805924152。[17] M. Hewitt、J. Ortmann 和 W. Rei。面向不确定性下决策的基于决策的情景聚类。*Annals of Operations Research*，315(2)：747–771，2022 年 8 月。doi：https://doi.org/10.1007/s10479-020-03843-x。[18] J. L. Higle。随机规划：不确定性重要时的优化，第 30–53 页。INFORMS，2005 年 9 月。ISBN 9781877640216。doi：http://dx.doi.org/10.1287/educ.1053.0016。

[19] K. Kim 和 V. Zavala。用于随机混合整数规划的对偶分解方法的算法创新与软件。10:225–266，2017年11月。doi：https://doi.org/10.1007/s12532-017-0128-z。[20] G. Laporte 和 F. V. Louveaux。用于完全补救随机整数规划的整数L形方法。《运筹学快报》，13 142，1993 3 133 0167 6377 10 1016 0167( ): – 。ISSN - 。doi：https://doi.org/。/ - 6377(93)90002-X。[21] T. L. Magnanti 和 R. T. Wong。加速 Benders 分解：算法改进与模型选择准则。《运筹学》，29(3)：464–484，1981。ISSN 0030364X，15265463。doi：https://doi.org/10.1287/opre.29.3.464。[22] B. Markhorst、M. Leitner、J. Berkhout、A. Zocca 和 R. D. van der Mei。用于求解大规模随机混合整数规划的两步热启动方法。2024。doi：https://doi.org/10.48550/arXiv.2412.10098。[23] B. S. Narum、J. Fairbrother 和 S. W. Wallace。通过分解输出分布进行基于问题的情景生成。《欧洲运筹学杂志》，318(1)：154–166，2024。[24] A. Papavasiliou 和 S. S. Oren。输电受限网络中高风电渗透率下的多区域随机机组组合。《运筹学》，61 592，2013 3 578。doi：https://doi.org/。/opre. . .[25] B. Polyak。求解极值问题的一般方法。《苏联数学文摘》，8：33–36，1967年1月。[26] R. Rahmaniani、T. G. Crainic、M. Gendreau 和 W. Rei。Benders 分解算法：文献综述。《欧洲运筹学杂志》，259 817，2017 3 801。[27] R. T. Rockafellar 和 R. J.-B. Wets。不确定性优化中的情景与策略聚合。《运筹学数学》，16：119–147，1991 年2月。doi：https://doi.org/10.1287/moor.16.1.119。[28] J. J. Torres、C. Li、R. M. Apap 和 I. E. Grossmann。线性与混合整数两阶段随机规划软件性能综述。*Algorithms*，15(4)，2022。ISSN 1999-4893。doi：https://doi.org/10.3390/a15040103。[29] S. Wallace 和 W. Ziemba。随机规划的应用。5，2005年1月。doi：https://doi.org/10.1137/1.9780898718799。[30] R. F. Woolson。*Wilcoxon* 符号秩检验，第1–3页。John Wiley & Sons, Ltd，2008。ISBN 9780471462422。doi：https://doi.org/10.1002/9780471462422.eoct979。[31] X. Zhao 和 P. Luh。求解拉格朗日松弛对偶问题的新束方法1。优化理论与应用杂志，113：373–397，2002年5月。doi：https://doi.org/10.1023/A:1014839227049。

附录 A。步骤 3 详情（次梯度上升）

初始化（冷启动）。由于没有可用的先验信息，我们采用冷启动来初始化乘子： $\lambda^0{}_{(s)} = 0$ 对于所有 $s \in \mathcal{S}$。

步长规则。 为确定迭代 k 时的步长 α_k ，我们采用 Polyak 步长规则 [25]:

$$\alpha_k = \gamma_k \frac{Z_P^{\mathcal{S}'} - Z_D^{\mathcal{S}'}(\lambda^k)}{\|g^k\|^2}, \tag{A.1}$$

其中， $Z_D^{\mathcal{S}'}(\lambda^k)$ 是当前对偶目标值， $Z_P^{\mathcal{S}'}$ 是迄今为止发现的最佳上界（原始可行解），而 $\gamma_k \in (0, 2]$ 是一个初始化为 $\gamma_0 = 1.8$ 的缩放参数。

原始启发式（信赖域）。 为收紧 Polyak 规则所需的上界 $Z_P^{\mathcal{S}'}$ ，我们每 20 次迭代运行一次原始启发式。我们采用 固定并优化 策略：通过在当前最佳第一阶段解 $\bar{x}$ 周围施加 5% 容差邻域（信赖域），定义一个受限主问题（RMP）。这显著缩小了搜索空间，使混合整数规划求解器能够快速识别高质量可行解。

自适应调整与终止。 我们实施了一种用于抑制振荡的自适应机制：如果最佳对偶界连续五次迭代未得到改善，则更新 $\gamma_{k+1} \leftarrow 0.5\gamma_k$。 当达到最大运行时间 $T_{\max}$ 或相对最优性间隙低于 $\epsilon = 0.01\%$ 时，算法终止。

附录 B. 案例研究

模型公式化

为了通过实证评估 LOTUS 的性能，我们利用 [18] 中介绍的生产计划问题设计了一项全面的计算研究。该问题涉及一家制造商，该制造商使用有限的资源（如木材、表面处理和木工）生产家具（如书桌、桌子和椅子），且需求已知。我们将市场需求视为不确定参数，从而把该问题扩展到随机情境。我们还扩展了 [18] 的基本模型，以纳入若干反映现实运营约束的特征。虽然标准求解器仍能处理基本模型，但这些扩展带来了显著的计算挑战。具体而言，固定成本对应的二元变量、生产批次对应的整数性约束，以及服务未达标对应的惩罚逻辑，共同形成了一个高度非凸的搜索空间，传统分支定界方法难以有效遍历。我们得到如下 DEP 模型:

$$\min_{\bar{x}, \{y_{(s)}\}_{s\in\mathcal{S}}, \{\upsilon_{(s)}\}_{s\in\mathcal{S}}, \alpha} \quad c^T\bar{x} - \sum_{s\in\mathcal{S}} p_{(s)} q^T y_{(s)} + \sum_{s\in\mathcal{S}} p_{(s)} f^T \upsilon_{(s)} + u^T\bar{\alpha} \tag{B.1a}$$

$$\text{s.t.} \quad x_{(s)} \geq W y_{(s)}, \qquad \forall s \in \mathcal{S} \tag{B.1b}$$

$$y_{(s)} + \upsilon_{(s)} = d_{(s)}, \qquad \forall s \in \mathcal{S} \tag{B.1c}$$

$$x_{(s)} \leq \bar{x}, \qquad \forall s \in \mathcal{S} \tag{B.1d}$$

$$\alpha_{(s)} \leq \bar{\alpha}, \qquad \forall s \in \mathcal{S} \tag{B.1e}$$

$$\bar{x} \leq L^T\alpha, \tag{B.1f}$$

$$y_{(s)} \geq b\beta_{(s)}, \qquad \forall s \in \mathcal{S} \tag{B.1g}$$

$$y_{(s)} \leq d_{(s)}\beta_{(s)}, \qquad \forall s \in \mathcal{S} \tag{B.1h}$$

$$x_{(s)} \geq 0, \qquad \forall s \in \mathcal{S} \tag{B.1i}$$

$$\bar{x} \geq 0, \tag{B.1j}$$

$$y_{(s)} \in \mathbb{Z}_+^{|F|}, \qquad \forall s \in \mathcal{S} \tag{B.1k}$$

$$\upsilon_{(s)} \in \mathbb{Z}_+^{|F|}, \qquad \forall s \in \mathcal{S} \tag{B.1l}$$

$$\alpha \in \{0,1\}^{|R|}, \tag{B.1m}$$

$$\bar{\alpha} \in \{0,1\}^{|R|}, \tag{B.1n}$$

$$\beta_{(s)} \in \{0,1\}^{|F|}, \qquad \forall s \in \mathcal{S}. \tag{B.1o}$$

我们使用有限情景集合 $s \in \mathcal{S}$ 对不确定性进行建模。每个情景代表一种可能的未来需求状态 $d_{(s)}$，并被赋予概率 $p_{(s)}$，满足 $\sum_{s\in\mathcal{S}} p_{(s)} = 1$。模型中的决策变量包括获取的资源向量 $\bar{x} \in \mathrm{R}_+^{|R|}$ 、家具产量向量 $y_s \in \mathrm{Z}_+^{|F|}$, $\forall s \in \mathcal{S}$，以及未满足需求向量 $\upsilon_s \in \mathrm{Z}_+^{|F|}$, $\forall s \in \mathcal{S}$ 1 。

目标函数（B. a）旨在最小化所有情景下的净成本，即对所有情景求和，计算资源可变成本总额 $c^T\bar{x}$ 减去销售总收入 $q^T y_s$ ，再加上资源固定成本总额 $u^T\alpha$ 和未满足需求的预期惩罚 $f^T\upsilon_{(s)}$。该模型依赖若干关键参数。向量 $c \in \mathrm{R}_+^{|R|}$ 和 $q \in \mathrm{R}_+^{|F|}$ 分别包含获取各类资源的成本以及各类家具的销售价格。市场限制通过需求向量 $d_s \in \mathrm{R}_+^{|F|}$, $\forall s \in \mathcal{S}$ 表示，这些向量规定了每种情景下各类产品的最大可售数量。生产过程本身由技术矩阵 $W \in \mathrm{R}_+^{|R|\times|F|}$ 定义，其中每个元素 w_{rf} 表示生产一件家具 f 所需的资源 r 数量。约束（B.1c）定义了服务短缺。约束（B.1f）对资源使用量施加限制，并将资源获取与固定成本变量 α 联系起来。约束（B.1g）和（B.1h）共同处理生产的最小批量逻辑。

实现说明。 尽管第3节中的通用框架采用等式非预见性约束（ $x_{(s)} = x$） ，但为体现资源投资的实际情况，我们将情景间的横向耦合具体化为 $x_{(s)} \leq x$ 和 $\alpha_{(s)} \leq \alpha$ 。在生产计划中，第一阶段决策是沉没成本产能购置（例如机械设备或劳动力），其作用是设定上限；如果情景特定需求较低，强制耗尽资源在实际中并不合理。

从数学上看，这将对偶乘子限制为非负（ $\lambda \in \mathbb{R}_+^{|\mathcal{S}|}$ ）。虽然数值上更加稳定，但对偶函数仍不可微，这是其结构性特征，也会引发渐进停滞效应。在对偶分解中，这种效应会导致主问题的搜索停滞，使第一阶段候选解收敛过慢，无法得到高质量的原始解。我们的热启动策略通过将搜索直接转入共识度较高的区域，绕过这一停滞阶段。因此，这些实验构成了结构性概念验证，表明即使对偶界呈现次梯度法典型的缓慢渐近收敛，该方法仍能识别高质量的原始解。

实例生成

为全面测试 LOTUS，我们需要一组计算难度较高的问题实例，并且这些实例须符合第3.2节所述的理论要求和实际动机。我们的生成流程通过系统性地设计具有块角结构的实例，并赋予其已知会导致弱线性规划松弛的结构特性，从而形成较大的整数性间隙；这正是采用拉格朗日分解而非标准分支定界求解器的动机。为满足紧致性和相对完全补救这两项理论要求，所有生产能力和库存上限均设为有限上界，并允许以较高罚成本出现未满足需求。生成方法结合了三个核心复杂性来源：

1. 固定成本结构：纳入较高的设置成本，以形成高度非凸的搜索空间。这直接检验 LOTUS 在 $Z_{LP}^{\mathcal{S}}$ 提供的引导效果不佳时缩小整性间隙的能力。2. 资源稀缺：设计了一个包含密集生产约束且容量紧张的矩阵，使得寻找可行原始界在计算上十分困难，因此需要 LOTUS 热启动提供的稳定乘子引导。3. 场景异质性：生成高方差的随机需求场景，以构造复杂的对偶空间。这评估了该框架的主要实际动机：通过计算成本较低的子集识别具有代表性的活动流形。

根据初步实验，我们将热启动配置固定为在 30% 的子集上迭代 10 次。在一系列主要实验中，我们采用此热启动配置，在广泛的问题实例上对 LOTUS 与标准 DD 算法 [5] 进行直接比较（5 个随机种子，每个种子 24 个实例）。所有实例均按照紧致性和相对完全补救的理论假设生成，以确保次梯度更新的稳定性以及原始恢复的可行性。

与工业界的利润最大化目标一致，实验采用生产计划问题的最大化形式。在这一背景下，拉格朗日对偶界作为理论上界，而投影启发式找到的可行解则构成原始下界。通过这种设置，我们能够明确衡量 LOTUS 通过加速原始收敛来改善下界的能力，即使在上界表现出典型渐进停滞效应的情景中也是如此。

所有实验均在一台配备 8GB 内存的 2.4GHz CPU 集群上以单线程运行，并使用 Python 3.9 中的 Gurobi 10.0 [15] 。用于生成生产计划问题实例和运行实验的代码可在 GitHub [9] 上找到。

附录C：进一步探讨速度增益

速度增益的理论洞见

通过分析计算成本与信息获取之间的权衡，可以将加速原始值收敛的理论有效性形式化。设 $t_L(\mathcal{S})$ 表示在场景集 $\mathcal{S}$ 上执行一次次梯度迭代所需的平均时间。与对偶分解相比，LOTUS 的计算优势依赖于以下两个结构性特征：

1. 成本不对称性。次梯度迭代的计算成本随情景数线性增长。因此，在缩减集 $\mathcal{S}'$ 上执行的迭代，成本显著低于在完整场景集上执行的迭代：

$$t_L(\mathcal{S}') \ll t_L(\mathcal{S}). \tag{C.1}$$

这种不对称性使 LOTUS 能够以较低成本完成对偶空间所需的“探索”。

2. 活跃集识别。 速度增益在很大程度上取决于约化问题与完整问题之间信息传递的质量。设 $\mathcal{A}_{\mathcal{S}}$ 为完整问题最优解中的活跃非预见性约束集合（即乘子 λ非零的索引）。类似地，设 $\mathcal{A}_{\mathcal{S}'}$ 为约化问题识别出的活动集。当重叠度 $|\mathcal{A}_{\mathcal{S}} \cap \mathcal{A}_{\mathcal{S}'}|$ 较大时，净速度增益最大。如果缩减集 $\mathcal{S}'$ 具有代表性，热启动 λ^{WS} 就能正确预测 $\mathcal{A}_{\mathcal{S}}$中的很大一部分。

因此，求解完整问题所需的高成本迭代次数 N_{full} 会减少，减少的数量相当于此前已执行的低成本迭代次数 N_{red} 。渐近加速收益可近似表示为：

$$\mathrm{Gain} \propto \underbrace{N_{\mathrm{red}} \cdot (t_L(\mathcal{S}) - t_L(\mathcal{S}'))}_{\substack{\text{Savings from} \\ \text{cheap exploration}}} - \underbrace{\mathrm{Loss}(\mathcal{S}, \mathcal{S}')}_{\substack{\text{Correction for} \\ \text{misidentified active sets}}}. \tag{C.2}$$

该公式表明，当成本差异 $t_L(\mathcal{S}) - t_L(\mathcal{S}')$较大，且完整问题与缩减子问题之间的结构重叠度较高时，LOTUS 的效果最佳，从而最大限度地减少热启动阶段将乘子错误识别为活跃或非活跃后进行纠正所产生的“损失”。

加速收益的数值结果

表 C.3 对比了每次迭代的平均时长，量化了 LOTUS 相较于对偶分解的效率提升。

表 C.3：LOTUS 热启动阶段 $t_L(\mathcal{S}')$、LOTUS 主阶段 $t_L(\mathcal{S}')$ 及对偶分解 $t_{DD}(\mathcal{S}')$中每次次梯度上升迭代的平均用时（秒）。

	LOTUS		DD	速度增益
$\lvert\mathcal{S}\rvert$	$t_L(\mathcal{S}')$（秒）	$t_L(\mathcal{S})$（秒）	$t_{DD}(\mathcal{S})$（秒）	(%)
250	17.2	23.5	27.2	13.6
500	29.2	47.6	56.3	15.4
1000	50.2	89.3	104.3	14.4
2000	72.0	162.0	200.0	19.0

在 $|\mathcal{S}| = 2000$，在缩减集上进行一次迭代仅需 72.0 秒，而一次对偶分解迭代则需 200.0 秒。这一成本降低了 2.7 倍，使 LOTUS 能够在成本显著更低的环境中进行初始对偶空间探索。有趣的是，即使在求解全场景集时，LOTUS 也展现出更快的速度。在 LOTUS 的步骤 5 中，平均迭代时间为 $162.0s$，比对偶分解快 19%，其平均迭代时间为 $200.0s$。这表明，热启动乘子 λ^{WS} 能够为解空间提供高质量的初始协调，从而简化求解完整问题的过程。随着场景数减少，LOTUS 迭代相对于对偶分解迭代的加速比

表 C.4：LOTUS 和对偶分解的迭代时间预算分配。报告的时间和迭代次数为所有实例的平均值（对于对偶分解的完整阶段，我们报告完整运行时间 7200 秒）。

	步骤 3		步骤 5		DD
$\|\mathcal{S}\|$	时间（秒）	迭代次数	时间（秒）	# 次迭代	# 次迭代
250	172	10	7028	299	265
500	292	10	6908	145	128
1000	502	10	6698	75	69
2000	720	10	6480	40	36

略有下降。总体而言，LOTUS 主阶段中的迭代比对偶分解中的迭代快 15.8%。

为评估 LOTUS 热启动策略的影响，我们分析计算开销和迭代效率随场景数（$|\mathcal{S}|$）的变化情况。与对偶分解相比，LOTUS 性能优势的核心在于成本不对称性，见附录 C 第 3 节：第 3 步（求解热启动）使用缩减后的场景集（$\mathcal{S}'$），而第 5 步（求解完整问题）使用全场景集（$\mathcal{S}$）。表 C.4 总结了 LOTUS 和对偶分解如何分配 120 分钟（$7200s$）的运行时间预算。平均而言，LOTUS 的热启动阶段（第 3 步）耗时 415 秒，约占总运行时间预算的 5.76%。随着场景数增加，热启动的绝对耗时也随之增长；当 $|\mathcal{S}| = 2000$ 时，耗时达到 720 秒（占运行时间预算的 10%）。