arXiv:2303.06724v1 [math.OC] 12 Mar 2023

¨用于计算机会成本矩阵的格罗布纳基与格雷弗基

YUCHEN GE、JANOSCH ORTMANN 和 WALTER REI

Abstract. 机会成本矩阵最早由 [15], 提出，在情景约简的背景下具有重要意义。我们基于代数几何中的思想，提出了新的算法，利用计算代数几何高效计算机会成本矩阵。我们通过为两个随机整数规划问题计算机会成本矩阵，展示了这些算法的有效性。

1. Introduction

在许多现实世界的应用中，不确定性下的决策是一个具有挑战性的问题，需要考虑多个可能的情景。然而，情景数量可能随着不确定参数数量的增加而呈指数增长，从而使决策问题变得难以处理。情景约简方法旨在通过减少情景数量，同时保留原问题的关键特征，来解决这一问题。

最近， [1, 15, 16]问题驱动的情景约简方法取得了进展。所有这些方法都需要计算 [15]中介绍的<样式 id='2'>机会成本矩阵的某种变体：为每个情景s_j分配一个决策 x_j^*。在 [15, 16] 中，这是通过为单情景问题选择一个最优解来实现的。机会成本矩阵通过将与情景 j 相关的解在情景 i下进行评估来计算。直观地说，假设我们有一个预言机，它预测情景 j 将会发生。机会成本矩阵的 (i, j) 项表示：相信预言机并根据情景 j 的视角采取决策后，如果最终事实证明实际发生的是情景 i ，所产生的成本。

机会成本矩阵自然导出了情景集上的基于决策的距离：如果机会成本矩阵的 (i, j) 项较小（因此，在实际情景 i 发生时预测情景 j 所导致的成本也较小），则可以认为两个情景 i 和 j 彼此接近；而如果该矩阵的 (i, j) 项较大，则可以认为二者相距较远。 [15, 16] 提出了基于这一距离的聚类算法，结果表明，这些算法能够得到良好的近似解，并为真实目标值提供紧的界。

在本文中，我们提出了一种利用 Gr¨obner 基与 Graver 基高效计算机会成本矩阵的新方法。Gr¨obner [5] 基与 Graver [11] 基是代数对象

2000 *Mathematics Subject Classification.* 90C15（主要分类）90C10、13P25（次要分类）。本研究是在 YG 获得资助于蒙特利尔担任研究实习生期间构思的，资金由 MITACS Globalink计划提供。作者感谢 MITACS 提供这项支持。JO 谨向 R¨udiger Schultz 致谢，感谢他就本文主题所进行的宝贵讨论。他衷心感谢 NSERC发现基金提供的支持。在撰写本文期间，第三作者担任加拿大研究讲席（CRC）随机运输与物流系统优化讲席。此外，他还通过发现基金项目获得了加拿大自然科学与工程研究委员会（NSERC）的支持。他衷心感谢 CRC 和 NSERC 项目提供的所有支持。

可用于求解多项式方程与不等式系统。这些方法已成功应用于优化 [19, 20] ，并已广泛应用于各个领域

我们的方法利用机会成本矩阵的结构，将其构造为多项式系统，随后可使用 Gr¨obner 基和 Graver 基求解。所得方法速度快、精度高，并且能够随着情景数和变量数的增加而扩展。

我们通过对来自 [13] 的一个知名整数规划进行数值实验，展示了该方法的有效性。结果表明，当情景数量较多时，Gr¨obner 基方法尤其高效；而当决策变量数量较多时，Graver 基方法仍能保持稳定。本文为机会成本矩阵的计算提供了两种新方法，从而对相关文献作出了贡献。我们还证明了关于确定性和随机整数规划中 Graver 基的理论结果。

本文其余部分的组织结构如下。在第 2 节中，我们介绍机会成本矩阵及其用途的相关背景。在第3 节中，我们引入 Gr¨obner 基和 Graver 基，并给出我们的数学结果。第4 节介绍随后用于生成第 5 节数值结果的算法。最后，第 6 节作出总结。

记号。 在本文中， $\mathbb{Z}$表示整数集， $\mathbb{Z}_{\geqslant 0}$ 表示非负整数的子集。

2. Opportunity cost matrices

考虑以下两阶段随机优化问题

$$\min_{x\in\mathcal{X}} \mathbb{E}F(x,\xi) = \min_{x\in\mathcal{X}} \gamma^\top x + \mathbb{E}Q(x,\xi) \tag{2.1}$$

其中， $\gamma \in \mathbb{R}^d$ ，而 $\mathcal{X}$ 是整数格与凸多面体的交集。目标函数的随机部分 Q 由下式给出

$$Q(x,\xi) = \min\left\{c_\xi^\top y \colon Tx + Ay = h_\xi \colon y \in \mathbb{Z}^d\right\} \tag{2.2}$$

其中，对于所有 ξ 和 T， $c_\xi \in \mathbb{Z}^d$ 和 $b_\xi \in \mathbb{Z}^k$ 是相应维度的整数矩阵。注意，我们假设 T 和 A 是确定性矩阵，即它们不是随机的。

一般而言，如何获得（2.1）的显式解并不明显。随机规划中的情景方法 [4] 包括用离散测度来近似概率测度 $\mathbb{P}$在 ξ 上诱导出期望算子 $\mathbb{E}$ ，该离散测度为[1]p_1，…， p_N在有限结果集 ξ_1，…，ξ_N（通常称为情景）上分配概率，其中 ξ。集合 $\mathcal{S}$ ={ξ_1，…ξ_N} 随后被称为情景集。

例如，可以通过样本均值近似 [17], 实现，其中 ξ_i是从 $\mathbb{P}$ 中独立抽取的样本，并且每个样本都被赋予相同的发生概率 $1/N$ 。无论如何，情景方法都会产生如下形式的优化问题

$$\min\left\{\gamma^\top x + \sum_{i=1}^N p_i c_i^\top y_i : x \in \mathcal{X}, Ay_i = h_i - Tx, y_i \in \mathbb{Z}_+^n \ \forall\, i\right\}. \tag{2.3}$$

为了恰当地建模 $\mathbb{P}$所表示的不确定性，通常必须生成大量情景。另一方面，当2.3中的 N 较大时，求解这类问题可能会带来极大的计算挑战。这促使我们思考情景约简：能否

[1]即 $p_n \geqslant 0$ 对于所有 n 和 $\sum_{n=1}^N p_n = 1$

ˆ ˆ找到一个小的子集 $\mathcal{S}$，它属于情景集 $\mathcal{S}=\{\xi_1,\ \ldots\ldots,\ \xi_N\}$ 使得将 $\mathcal{S}$ 替换为 $\mathcal{S}$ 在（2.3）中仅产生较小的近似误差。

一般而言，情景约简方法可以分为基于分布的和基于问题的方法。前者旨在最小化子集的经验测度与原始情景集之间的差异（以某种适当的方式定义）。例如，参见 [8, 14, 18]，其中给出了这类方法的示例。另一方面，基于问题的方法 [1, 15, 16] 旨在最小化规模较大的问题与规模较小的问题所对应解之间的差异，换言之，如果情景根据（2.3）导致相似的决策，则认为这些情景彼此相似。在 [15], 中介绍了机会成本矩阵。

定义 2.1。 考虑一组可行决策 $x_1,\ \ldots\ldots,\ x_N\in\mathcal{X}$，其中决策 x_j 与情景 s_j 相关联。机会成本矩阵是 $N\times N$ 矩阵 $\mathcal{A}$，其 $(i,\ j)$元素为 $\mathcal{A}_{ij}=F(x_i,\ \xi_j)$。

显然，解 x_j 有许多有趣的选择。

例 2.2。 在 [15] 和 [16], 中，决策 x_n 被选为（2.1）的单情景解，即

$$x_j\in\operatorname{argmin}F(x,\xi_j), \tag{2.4}$$

本文旨在针对（2.1）中定义的随机线性两阶段问题，寻找高效计算矩阵 $\mathcal{A}$的方法。由于函数 Q 和 F 仅相差线性项 $\gamma\cdot x$，因此我们将注意力转向高效计算

$$Q\left(x_i,\xi_j\right)=\min\left\{c_j^{\top}y\colon Ay=b_{ij}\right\}, \tag{2.5}$$

其中 $c_j=c(\xi_j),\ \ b_{ij}=b(x_i,\ \xi_j)$。这里，我们记 $b(x,\ \xi)=h_\xi-T_\xi x$。由于我们假设 A是确定性的，（2.5）右侧的可行域只会随着 b_{ij} 而变化，当 i 和 j 发生变化时。

3. Gröbner and Graver bases

本节将发展我们所需的Gr¨obner和Graver基理论，以论证并说明第 4节将介绍的算法。

在上一节中我们看到，为了计算机会成本矩阵，我们需要求解形如

$$(IP)_{c,b}:\quad \min\left\{c^{\top}z:Az=b,z\in\mathbb{N}^d\right\} \tag{3.1}$$

给定整数矩阵 A 但成本向量 $c\in\mathcal{C}$ 和约束向量 $b\in\mathcal{B}$各不相同的 $\mathcal{C}$ 和 $\mathcal{B}$ 是分别对应于 $c(\xi_m)$ 和 $b(x,\ \xi_m)$ 在（2.5）中可取值的有限集合。由于我们始终将矩阵 A 视为固定矩阵，因此在记号中不将其列出。

注意，（3.1）的最优解可能不唯一，这有时会带来不便。因此，在需要时，我们可以用其细化形式 $(IP)_{>_c,\ b}$ 来替代（3.1），下面将对其进行定义。我们首先定义 [2] 上的一个全序 $\mathbb{Z}^d_{\geqslant 0}$。

定义 3.1。 令 $>$ 为 $\mathbb{Z}^n_{\geqslant 0}$ 上的任意全序，并固定 $c\in\mathbb{Z}^d$。我们通过如下方式在 $\mathbb{N}^n$ 上定义另一个序关系：规定 $x>_c$ 当且仅当 y

(1) $c\cdot x>c\cdot y$ or

[2] A *total or*$_{der}$ 在集合 S 上是一个二元关系 $\prec$ 在 S 上，使得对于所有 a $a\prec b$ 或 $b\prec a$，$b\in S$ 其中 $a\neq b$.

（2） $c \cdot x = c \cdot y$ 和 $x > y$。

除非另有说明， $\prec$ 定义3.1 中的全序将采用字典序，在该序下， $x > y$ 当且仅当 $x - y$ 的最左侧非零项为正。

定义3.2。集合 $\mathcal{T}_c \subseteq \mathbb{Z}^d$ 被称为问题族 $\{(IP)_{c,\, b}$ 的测试集： $b \in \mathbb{R}^l\}$ 当

(1) $c \cdot t > 0$ 对所有 $t \in \mathcal{T}_c$(2) 对每个 $b \in \mathbb{R}^l$ 以及每个关于 $Az = b$ 的非最优可行解 $z_0 \in \mathbb{Z}^d_+$ ，都存在一个向量 $t \in \mathcal{T}_c$ ，使得 $z_0 - t$ 是可行的。这样的向量称为增广向量。

给定测试集 $\mathcal{T}_c$，增广算法（算法1）提供了一种计算 $(IP)_{>c,\, b}$ 对任意 $b \in \mathcal{B}$的最优解的方法。直观地说，只要存在 $t \in \mathcal{T}_c$ ，满足 $c^\top t > 0$ 且 $z_0 - t$ 可行，我们就反复迭代 z_0，令 $z_0 := z_0 - t$。因此，我们希望为每个 $c \in \mathcal{C}$高效地构造测试集 $\mathcal{T}_c$ 。注意，由于对于任意 $t \in \mathcal{T}_c$ ， z_0和 $z_0 - t$ 都是可行的，这意味着 $\mathcal{T}_c \subseteq \ker(A)$ ，因此

$$At = A(z_0 - (z_0 - t)) = Az_0 - A(z_0 - t) = b - b = 0. \tag{3.2}$$

下面我们将介绍构造此类测试集的两种方法。第一种方法使用Gr¨obner基，将在第3.1节中介绍。第二种方法基于Graver基，更进一步构造一个通用测试集，即一个 $\mathcal{T}$ 包含测试集的集合， $\mathcal{T}_c$其中每个成本向量 $c \in \mathcal{C}$对应一个测试集。

3.1. **Gr¨obner基方法**。现在众所周知， [2, 19, 20, 23]Gr¨obner基 [5, 7] 可以生成测试集。Buchberger算法 [5, 6] 以一组多项式作为输入f_1，……， f_M并返回由¨生成的理想的约化Gr $f_j$obner基。Buchberger算法的一个几何版本见于 [24]。当生成集较小时，该算法的效率最高。我们通过使用环面理想，特别是采用 [3]中的方法来实现这一点。其优势在于，对于所有成本向量，只需执行一次即可完成，这对于计算机会成本矩阵的场景而言非常高效。

我们将需要以下结果，它是迪克森引理 [7]的直接推论，可在 [24]的引理 2.1.4中找到。

引理3.3。 设 $A \in \mathbb{Z}^{l \times d}$ ，并令$\mathcal{N}_{c,\, b}$ 表示$(IP)_{c,\, b}$的非最优可行解集合。那么存在 $\alpha(t)$、……、$\alpha(t) \in \mathbb{N}^d$，使得

$$\bigcup_{b \in \mathbb{R}^l} \mathcal{N}_{c,b} = \bigcup_{i=1}^{t} (\alpha(i) + N^n) \tag{3.3}$$

向量 $\alpha(1)$，……， $\alpha(t)$ 使我们能够利用以下结果为整数规划族$\{(IP)_{c,\, b}$ 计算测试集 $b \in \mathcal{B}\}$; 该结果由Gordon-Dickson引理推出，并在 [24]中表述为推论2.1.10。另见 [21]中的引理2.1。

定理3.4。 固定 $c \in \mathbb{Z}^d$，令 $\alpha(1)$，……， $\alpha(t)$ 如引理*3.3*中所述，并令 $\beta(i)$ 为规划$(IP)_{>c,\, A\alpha(i)}$的（唯一）最优解。那么集合

$$\mathcal{G}_{A,c} = \{(\alpha(i) - \beta(i)), i = 1, 2, ..., t\} \tag{3.4}$$

是问题族 $\{(IP)_{c,\, b}$： $b \in \mathbb{R}^l\}$ 的极小测试集。

定义 3.5。 集合 $\mathcal{G}_{A,\, c}$ 称为矩阵 A的 *Gr¨obner* 基。

虽然这一结果给出了所需测试集的公式，但它需要计算引理 $\alpha(i)$3.3 中的，更关键的是，还需要计算 $\beta(i)$，而后者本身是用整数问题的解给出的。文献 [24]给出了一种测试集的几何构造。

如上文所述，我们需要计算 $(IP)_{>_c,\ b}$ 在 c 变化时的解。如果我们希望采用格罗布纳基方法，就需要针对每个成本向量 $c \in \mathcal{C}$重新计算格罗布纳基。因此，我们采用一种不同的方法，该方法基于多项式环理论中的思想。

令 $R = \mathbb{R}[x_1, \ldots\ldots, x_n]$ 表示 n 元多项式环。我们将经常利用整数向量与 R中单项式之差之间的密切关系。任意由非负整数构成的 n向量 $\alpha = (\alpha_1, \ldots\ldots, \alpha_n) \in \mathbb{Z}^n_{\geqslant 0}$ 都可以与一个单项式 $x^\alpha \in R$ 对应，该单项式定义为

$$x^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n} = \prod_{j=1}^{n} x_j^{\alpha_j}, \tag{3.5}$$

并且可以按如下方式将这一映射扩展到整数值向量。对于任意 $\alpha = (\alpha_1, \ldots\ldots, \alpha_n)\mathbb{Z}^n$，定义 $\alpha^+ = (\alpha_1^+, \ldots\ldots, \alpha_n^+)$和 $\alpha^- = (\alpha_1-, \ldots\ldots, \alpha_n^-)$，回顾一下，实数 x 的正部定义为 $x^+ = \max(x, 0)$ 而 x 的负部为 $x^- = (-x)^+$。那么， α^+， $\alpha^- \in \mathbb{Z}^n_{\geqslant 0}$ 和 $\alpha = \alpha^+ - \alpha^-$均成立。现在我们定义一个映射

$$\phi\colon \mathbb{Z}^n \longrightarrow R \quad \text{by} \quad \phi(\alpha) = x^{\alpha^+} - x^{\alpha^-}. \tag{3.6}$$

与定义 ¨3.5 中矩阵相关的格¨罗布纳基¨概念实际上是一个更一般概念的特例，即多项式理想的格 [22]罗布纳基。下面我们简要介绍格¨罗布纳基的定义及基本性质。有关更多细节，请参见例如第 1 章 [22]。

定义 3.6。 若子集 $I \subseteq R$ 满足以下两个性质，则称其为 R 的理想：

a) 对任意 x， $y \in I$ 都有 $x - y \in I$ （也就是说， I 是 R的加法子群）， b) 对任意 $x \in I$ 和 $r \in R$ 都有 $rx \in I$。

例 3.7。 给定多项式 f_1，.. ., f_n，由 f_1，.. ., f_n 生成的理想是所有 R-线性组合的集合：

$$\langle f_1, \ldots, f_n \rangle = \left\{ \sum_{j=1}^{n} r_j f_j \colon r_1, \ldots, r_n \in R \right\}. \tag{3.7}$$

例 3.8。 矩阵 $A \in \mathbb{Z}^{d\times n}$ 的环面理想 I_A 由满足 $u - v$ 属于 A的核的单项式之差$x^u - x^v$ 生成：

$$I_A = \langle x^u - x^v \colon Au = Av \rangle. \tag{3.8}$$

请参阅 [22] 的第4节，以了解环面理想理论。多项式 $f \in R$ 由单项式组成x^α ，其次数为 $\alpha \in \mathbb{Z}^n_{\geqslant 0}$。在全文中，我们固定一个全序（回顾定义3.1 及其相关讨论） $\prec$ 于 $\mathbb{N}^n$之上。原则上，任何全序都可以，但对我们而言， $\prec$ 始终是定义 $<_c$ 中所定义的全序3.1。给定 $\prec$后，我们可以通过比较次数来比较不同的单项式。由此，我们可以将初始单项式定义为 f 中次数最大者（相对于 $\prec$而言）。 f 的初始单项式记作

$\text{in}_{\prec}(f)$。给定 R的理想 I， I的初始理想是由 I中元素的初始单项式生成的理想：

$$\text{in}_{\prec}(I) = \langle \text{in}_{\prec}(f) : f \in I \rangle . \tag{3.9}$$

现在我们可以定义多项式理想的 Gr¨obner 基。

定义 3.9。 设 I 是 R的一个理想。若 I 的初始理想由 $\mathcal{G}$中各元素的初始单项式生成，则称 $\mathcal{G}$ 为 I 的 *Gr¨obner* 基：

$$\text{in}_{\prec}(I) = \langle \text{in}_{\prec}(g) : g \in \mathcal{G} \rangle . \tag{3.10}$$

若从 $\mathcal{G}$中移除任意一个元素后，条件 （3.10）不再成立，则称 $\mathcal{G}$ 的 Gr¨obner 基为极小。此外，若对于任意不同的 g_1， $g_2 \in \mathcal{G}$ 中的任一项都不是 g_1 在$\prec(g_2)$中的倍数，则称 $\mathcal{G}$ 为约化的。

给定 R 的一个理想 I 以及 $\mathbb{N}^n$上的一个全序，只有一个约化 Gr¨obner 基 $\mathcal{G}$，使得对于每个 $g \in \mathcal{G}$，$\text{in}_{\prec}(g)$ 的系数都等于 1。换言之，约化 Gr¨obner 基除将各元素乘以常数外是唯一的。

命题3.10。 回忆 $\alpha(i)$ 和 $\beta(i)$ 来自 *3.4*的向量。多项式集合

$$\mathcal{G}_{I_A} = \left\{ x^{\alpha(j)} - x^{\beta(j)} : j \in \{1, \ldots, t\} \right\} \tag{3.11}$$

i是关于序 $<_c$的环面理想 I_A 的约化*Gr¨obner*基 .

我们得出结论，Gr¨obner基的两个概念（一方面是矩阵的Grobner基，另一方面是多项式理想的Grobner基）实际上是相同的：它们通过映射 ϕ 建立联系，该映射定义于（3.6）中。事实上，最初的Buchberger算法 [5]就是在这一背景下提出的。

在 [3], 中，给出了一种高效算法，用于计算环面理想 I_A的小生成集，见算法 2。注意，环面理想只取决于矩阵 A，而不取决于成本向量 c。这促使我们提出核算法，其详细描述见算法 3：

(1) 给定矩阵 A，根据 [3]计算环面理想 I_A 的小生成集。(2) 对每个 $j \in \{1, ..., N\}$，应用Buchberger算法，计算关于 A 和 $<_{c_j}$的Gr¨obner基。(3) 使用（3.6）中的对应关系 $(x^u - x^v) \longleftrightarrow u - v$，获得关于 $(A、 c_j)$的测试集。(4) 对每个 $i、 j \in \{1、 ...、 N\}$，应用增广算法（算法 1）计算 $Q(x_i、 \xi_j)$，从而得到机会成本矩阵的 $(i、 j)$ 项。

3.2. **Graver 基方法**。 在上一节中，计算 $N \times N$ 机会成本矩阵仍然需要通过 Gröbner 基计算 N 测试集。这引出了一个问题：是否可以构造一个单一的测试集，使其不仅涵盖所有右端向量 b，还涵盖所有成本向量 c。

定义 3.11。 如果集合 $\mathcal{T} \subseteq$ 对于矩阵 A 包含针对每个成本向量 c的测试集$\mathcal{T}_c$，则称其为该矩阵的通用测试集 。

显然，仅采用 Gröbner 基方法不足以计算通用测试集，因为这需要计算无穷多个 Gröbner 基。相反，我们将采用 Graver 基方法。我们首先给出 Graver 基的定义。

定义 3.12。 设 $a = (a_1、 a_2、 ....、 a_n)$、 $b = (b_1、 b_2、 ....、 b_n) \in \mathbb{Z}^n$。则 $a \sqsubseteq b$ 当 $\forall i$ 时：$a_i b_i \geqslant 0$且 $|a_i| \leqslant |b_i|$。

定义 3.13。 固定一个整数矩阵 A。 A 的 Graver 基是 $\mathrm{Ker}_{\mathbb{Z}}(A)$ 中关于 $\sqsubseteq$的极小元素集合: $= \{u \in \mathbb{Z}^n : \ A \cdot u = 0、\ u \neq 0\}$，记作 Γ_A。

并不能立即看出 Γ_A 是有限的。然而， Γ_A 的有限性可由以下从 [12]汇总的结果推出。

命题 3.14。 设 $\mathcal{G}_{A,\ c}$ 是关于 $> c$的 A 的约化格罗布纳基。那么

$$\bigcup_{c \in \mathbb{Z}^n} \mathcal{G}_{A,c} \subseteq \mathcal{GR}_A.$$

接下来，我们陈述并证明一个关于随机整数规划2.3 的格雷弗基与确定性整数规划的格雷弗基之间关系的结果；后者通过选择一个情景并为其赋予概率1而获得:

$$\min \left\{ c^\top x + pq^\top y : Ax = b, x \in \mathbb{Z}_+^m, Tx + Wy = h, y \in \mathbb{Z}_+^n \right\} \tag{3.12}$$

然后，我们提出一个新的结果，该结果建立了 SIP 问题的格雷弗基与整数规划问题的格雷弗基之间的联系:

定理 3.15。 将 *SIP* 问题 *(5)* 的格雷弗基记为 Γ_N，将整数规划问题 (3.12) 的格雷弗基记为 Γ_1，其中 $i = 1,\ 2,\ \ldots\ldots,\ N$。若 $\mathrm{Ker}_{\mathbb{R}}(A) = 0$，则有如下关系

$$\Gamma_N = \{(0, 0, ..., v_i, ..., 0) : (0, v_i) \in \Gamma_1, i = 1, 2, ..., n\}$$

证明。 （2.3）的对应系数矩阵具有如下形式

$$A_N := \begin{pmatrix} A & 0 & 0 & \cdots & 0 \\ T & W & 0 & \cdots & 0 \\ T & 0 & W & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ T & 0 & 0 & \cdots & W \end{pmatrix}$$

（6）的对应系数矩阵具有如下形式

$$A_1 := \begin{pmatrix} A & 0 \\ T & W \end{pmatrix}.$$

由于 $\mathrm{Ker}_{\mathbb{R}}(A) = 0$，我们有 $\mathrm{Ker}_{\mathbb{Z}}(A_N) = (0,\ v_1,\ v_2,\ \ldots\ldots v_n) : \ (0,\ v_i) \in \mathrm{Ker}_{\mathbb{Z}}(A_1)\}$。根据 Graver 基的定义，我们可以得出结论。□

4. Algorithms

本节将介绍由上一节的数学推导得出的数值算法。我们首先介绍增广算法，该算法利用测试集 $\mathcal{T}_c$ 和可行解 z_0，计算出 $(IP)_{c,\ b}$ 的最优解。

有关算法 2的详细信息，请参见 [3]。

核算法以矩阵 A、一组成本向量 c 和一个右端向量数组 b 为输入，先应用算法 2，再反复应用 Buchberger算法，从而计算出相应的机会成本矩阵。这样，便针对每个 A ，计算出相对于每个 $>_c$的 Gr¨obner 基。随后，增广算法 1 便给出每个 c 和每个 b所对应的机会成本矩阵元素。

对于 Graver 基方法，我们将修改后的增广算法 4 应用于以下形式的一族整数问题：min $\left\{c^\top x : \ Ax = b\right\}$ 其中 b和 c均可变。

算法 1 增广算法

要求： 成本向量 c、右端向量 b
要求： 矩阵 A
要求： 测试集 $\mathcal{T}_c$
要求： z_0，使得 $Az_0 = b$。
  当 $\exists t \in \mathcal{T}_c$ 使得 $z_0 - t$ 可行时 **执行**
    $z_0 \leftarrow z_0 - t$
  结束当
    返回 $z_0 =$ 的最优解 $\{c^T x: \quad Ax = b\}$

算法 2 计算环面理想的小生成集的算法 [3]

要求： 整数矩阵 A计算一个 $\mathbb{Z}$–基 K，用于 ker(π_*)寻找一个等价基 K'，使得 K' 的所有行都位于同一正交象限中。$J \leftarrow$ 负元素所在的所有列的指标集，并令 K'_J 为通过反转 J 所指标列的符号而从K' 得到的矩阵。$G_J \leftarrow \varphi(K'_J)$.**当** $J \neq \varnothing$ **时执行**取 $j \in J$，并令 $G_{J\setminus\{j\}}$ 为对关于消去 $x_j J \leftarrow J\setminus\{j\}$的项序作用于 $\langle G_J \rangle$ 的约化 Gr¨obner 基所得的结果。**结束循环 返回** I_A的生成集 $G_\varnothing$。

算法 3 核算法

需要： 成本向量 $\{c_1、\ldots\ldots、 c_N\}$
需要： 右端向量 $\{b_{i、j}\}_{i}^{N}{}_{、j=1}$
要求： 矩阵 A
  使用算法 {$f_1$2计算环面理想的生成集
  $i \leftarrow 1$
  当 $i \leqslant N$ **时执行**
    通过Buchberger算法计算 $(A, >_{c_i})$ 的Gr¨obner基 $\mathcal{G}_i$
    $j \leftarrow 1$
    当 $j \leqslant N$ **时执行**
      将$\mathcal{A}_{i,j} = (IP)_{c_i, b_{i,j}}$ 输入增广算法1进行计算 $\mathcal{G}_i$
      $j \leftarrow j + 1$
    结束循环
    $i \leftarrow i + 1$
  结束循环
    返回 机会成本矩阵 $(\mathcal{A}_{i、j})_{i}^{N}{}_{、j=1}$

算法 4 Graver基方法 要求：成本向量 $\{c_1$、……、 $c_N\}$要求：右端向量 $\{b_i$、 $_j\}_i^N$ 、 $_{j=1}$计算 A的 Graver 基 Γ_A 要求：$i \leftarrow 1$当 $i \leqslant N$ 时执行$j \leftarrow 1$当 $j \leqslant N$ 时执行 计算 $Ax = b_i$、$_j$的可行解 z_i、$_j$ 当 $\exists\, t \in \Gamma_A$： $c_i^\top t > 0$ 且满足 $z_0 - t$ 为可行时执行z_i、$_j$: $= z_i$、$_j - t$ 结束循环 计算 $\mathcal{A}_i$、$_j = c_j^\top z_i$、$_j$结束循环 结束循环 返回机会成本矩阵 $\mathcal{A}$

5. Preliminary results

本节将通过一些数值示例检验我们的方法。第一个是 [13]中介绍的一个广为人知的纯组合随机规划。第二个是一个随机网络设计问题。

5.1. 关于 [13]的示例。在 [13], 中，介绍并研究了以下随机整数规划。

$$\begin{aligned} \min \left\{ 35x_1 + 40x_2 + \frac{1}{N}\sum_{\nu=1}^{N} 16y_1^\nu + 19y_2^\nu + 47y_3^\nu + 54y_4^\nu \right\} \\ \text{s.t. } x_1 + y_1^\nu + y_3^\nu \geqslant \xi_1^\nu \\ x_2 + y_2^\nu + y_4^\nu \geqslant \xi_2^\nu \\ 2y_1^\nu + y_2^\nu \leqslant \xi_3^\nu \\ y_1^\nu + 2y_2^\nu \leqslant \xi_4^\nu, \\ x_1, x_2, y_1^\nu, y_2^\nu, y_3^\nu, y_4^\nu \in \mathbb{Z}_+ \end{aligned} \tag{5.1}$$

如 [13] 情景 ξ^ν 从四维方块中均匀选取 [300， 12000]×[300， 12000] × [200， 12000] × [200, 12000]。为了将问题转换为形式（2.1），我们为 u_j^ν 引入松弛变量 $j \in \{1$、2、3、 $4\}$ 并得到

$$\begin{aligned} \min \left\{ \frac{1}{N}\sum_{\nu=1}^{N} 35x_1 + 40x_2 + 16y_1^\nu + 19y_2^\nu + 47y_3^\nu + 54y_4^\nu \right\} \\ \text{s.t. } x_1 + y_1^\nu + y_3^\nu - u_1^\nu = \xi_1^\nu \\ x_2 + y_2^\nu + y_4^\nu - u_2^\nu = \xi_2^\nu \\ 2y_1^\nu + y_2^\nu + u_3^\nu = \xi_3^\nu \end{aligned} \tag{5.2}$$

$$y_1^\nu + 2y_2^\nu + u_4^\nu = \xi_4^\nu,$$
$$x_1, x_2, y_1^\nu, y_2^\nu, y_3^\nu, y_4^\nu, u_1^\nu, u_2^\nu, u_3^\nu, u_4^\nu \in \mathbb{Z}_+.$$

为了检验我们的方法，我们计算了程序在 Apple Silicon M1 芯片上运行所用的 CPU 时间（秒）。我们观察到，可以很容易地读出问题（5.2）的可行解

$$x_1 = x_2 = y_1^\nu = y_2^\nu = 0, y_3^\nu = \xi_1^\nu, y_4^\nu = \xi_2^\nu \quad (\nu = 1, \dots N) \tag{5.3}$$

本任务旨在比较上一节 SIP 问题中机会成本矩阵的计算成本。换言之，我们使用三种算法计算一系列整数规划问题。我们在 Apple Silicon M1 芯片上，利用 Python 的`time` 软件包测试程序以 CPU 秒数计的运行时间。此外，我们还引入了现代 MINLP 求解器 gekko，该求解器专为混合整数规划问题设计，由 Python 的`gekko` 软件包实现。我们将在两种情况下，将我们的算法 3和 4与 MINLP 求解器的计算成本进行比较。一种情况是固定子问题，另一种情况是固定情景数量。

我们将计算情景数量等于 1、20、40、60、80、100、120、140、160、180 和 200 时的计算时间。数值结果汇总于下表中。

TABLE 1. 不同求解器之间的比较（计时单位：秒）

场景数量 核方法		MINLP 求解器	Graver 方法
1	0.20098	11.33629	131.06186
20	1.21699	219.18796	131.66931
40	2.33655	439.71981	132.02384
60	3.53545	643.3165	132.35069
80	4.68832	994.41339	132.72208
100	5.93403	1250.31321	133.14721
120	7.24906	1508.7674	133.61075
140	8.56163	1803.57992	134.06991
160	10.08377	2075.8631	134.61703
180	11.25169	2333.33849	135.21214
200	12.88583	2584.11191	135.8798

值得注意的是，在所有情景中，计算 A 的环面理想的生成集仅耗时约 0.1 秒，这对 Gr¨obner 基的计算至关重要。相比之下，计算 A 的 Graver 基大约耗时 130 秒。此外，我们还在下方用折线图直观展示了该表格。横轴表示不断增加的情景数量，纵轴表示运行时间。图中三条不同颜色的曲线代表三种不同的方法。

图 5.1 显示，算法 4 初始化几乎耗时 130 秒，随后曲线缓慢上升。MINLP 求解器的计算成本增长极快，并在 4 时开始超过算法 $N = 20$。总体而言，与另外两种算法相比，算法 3 的曲线显示出更优的性能。

接下来，我们比较子问题复杂度增加时的情况。下面的图展示了情景规模固定为 200、但情景数量增加时计算成本的变化趋势。在图 5.2中，在情景规模固定为 200 的情况下，var*x 表示特定规划问题中的 x个变量。由曲线对比可以看出，

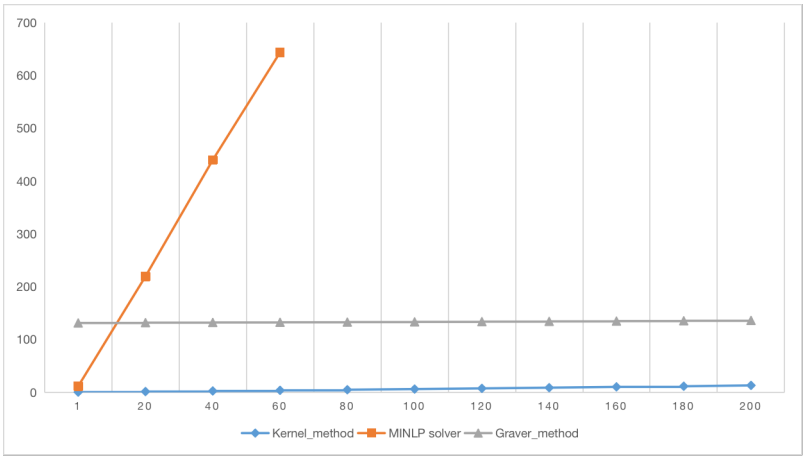


FIGURE 5.1. 子问题固定时计算成本的变化趋势

Gr¨obner 基方法（蓝色曲线）总体上优于 Graver 基方法（灰色曲线）和传统的 MINLP 求解器（黄色曲线）。此外，就细节而言，当问题复杂度达到 20 个变量时，Graver 基方法开始受到维数灾难的影响，而 Gr¨obner 基方法也出现了同样的情况，且程度更为严重。然而，MINLP 求解器在受到维数灾难影响时，并未反映出这种明显的复杂度。一般而言，MINLP 求解器适用于子问题规模较大、情景数量较少的情况。相比之下，Gr¨obner 基方法适用于子问题复杂度较低、情景数量较多的情况。

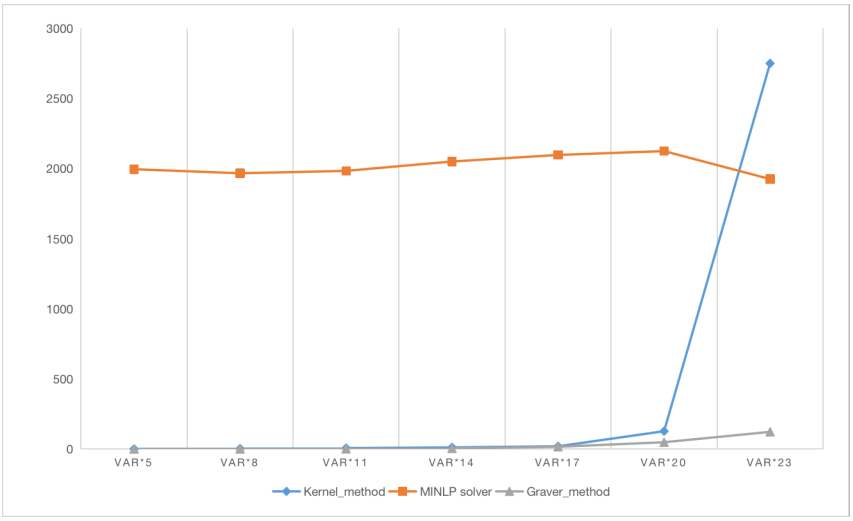


FIGURE 5.2. 情景数量固定时计算成本的变化趋势

我们在实验中还有一点尚未反映出来。通过数百次测试我们的算法，我们得出了经验性结论：我们提出的

算法在计算格罗布纳基时，针对 $N > 15$的耗时将大幅增加。而当复杂度介于 10 至 15 个变量之间时，计算 Graver 基所需的时间也会呈几何级增长。此外，令人惊讶的是，在超过 15 个变量的复杂度下，我们将无法再计算 Gr¨obner 基。我们将这一现象称为关于 SIP 问题的计算代数几何中的维数灾难。我们推测，其关键原因在于，随着维度增加，理想的最小生成元数量将大幅增加，从而使 Bunchberger 算法的计算成本变得高昂。

最后，问题的核心在于：当变量数量超过 15 个时，两种代数方法都无法运行，而计算某些特定理想的 Gr¨obner 基，具体而言即系数矩阵 A 的环面理想，是这两种算法中每一种都不可避免的步骤。我们尝试了许多专门用于代数计算的应用程序，例如 SageMath、Macaulay2 等。然而，当变量数量超过 15 个时，多项式理想的生成元的指数通常过大，导致算法所需的计算无法完成。例如，计算矩阵 $A = (B,\ I)$ 的 Gr¨obner 基时，其中 I 表示单位矩阵，而

$$B = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \\ 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \\ 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \end{pmatrix}$$

涉及对下方环面理想的 Gr¨obner 基进行计算

理想$(x_8^2 - x_7 * x_9,\ x_7 * x_8 - x_6 * x_9, x_6 * x_8 - x_5 * x_9, x_5 * x_8 - x_4 * x_9, x_4 * x_8 - x_3 * x_9,\ x_3 * x_8 - x_2 * x_9,\ x_2 * x_8 - x_1 * x_9,$
$x_1 * x_8 - x_0 * x_9,\ x_7^2 - x_5 * x_9,\ x_6 * x_7 - x_4 * x_9, x_5 * x_7 - x_3 * x_9, x_4 * x_7 - x_2 * x_9, x_3 * x_7 - x_1 * x_9,\ x_2 * x_7 - x_0 * x_9,$
$x_1 * x_7 - x_0 * x_8, x_6^2 - x_3 * x_9,\ x_5 * x_6 - x_2 * x_9, x_4 * x_6 - x_1 * x_9, x_3 * x_6 - x_0 * x_9, x_2 * x_6 - x_0 * x_8,\ x_1 * x_6 - x_0 * x_7,$
$x_5^2 - x_1 * x_9,\ x_4 * x_5 - x_0 * x_9, x_3 * x_5 - x_0 * x_8, x_2 * x_5 - x_0 * x_7, x_1 * x_5 - x_0 * x_6, x_4^2 - x_0 * x_8,\ x_3 * x_4 - x_0 * x_7,$
$x_2 * x_4 - x_0 * x_6, x_1 * x_4 - x_0 * x_5, x_3^2 - x_0 * x_6,\ x_2 * x_3 - x_0 * x_5, x_1 * x_3 - x_0 * x_4, x_2^2 - x_0 * x_4,\ x_1 * x_2 - x_0 * x_3, x_1^2 - x_0 * x_2,$
$x_8 * x_{11} * x_{12} * x_{13} * x_{14} * x_{15} * x_{16} - x_9, x_7 * x_{11} * x_{12} * x_{13} * x_{14} * x_{15} * x_{16} - x_8, x_6 * x_{11} * x_{12} * x_{13} * x_{14} * x_{15} * x_{16} - x_7,$
$x_5 * x_{11} * x_{12} * x_{13} * x_{14} * x_{15} * x_{16} - x_6, x_4 * x_{11} * x_{12} * x_{13} * x_{14} * x_{15} * x_{16} - x_5, x_3 * x_{11} * x_{12} * x_{13} * x_{14} * x_{15} * x_{16} - x_4,$
$x_2 * x_{11} * x_{12} * x_{13} * x_{14} * x_{15} * x_{16} - x_3, x_1 * x_{11} * x_{12} * x_{13} * x_{14} * x_{15} * x_{16} - x_2, x_0 * x_{11} * x_{12} * x_{13} * x_{14} * x_{15} * x_{16} - x_1,$
$x_9 * x_{10} * x_{14} * x_{15}^2 * x_{16}^3 - x_0 * x_6 * x_{11}^2 * x_{12}, x_8 * x_{10} * x_{14} * x_{15}^2 * x_{16}^3 - x_0 * x_5 * x_{11}^2 * x_{12},$
$x_7 * x_{10} * x_{14} * x_{15}^2 * x_{16}^3 - x_0 * x_4 * x_{11}^2 * x_{12}, x_6 * x_{10} * x_{14} * x_{15}^2 * x_{16}^3 - x_0 * x_3 * x_{11}^2 * x_{12},$
$x_5 * x_{10} * x_{14} * x_{15}^2 * x_{16}^3 - x_0 * x_2 * x_{11}^2 * x_{12}, x_4 * x_{10} * x_{14} * x_{15}^2 * x_{16}^3 - x_0 * x_1 * x_{11}^2 * x_{12}, x_3 * x_{10} * x_{14} * x_{15}^2 * x_{16}^3 - x_0^2 * x_{11}^2 * x_{12},$
$x_0 * x_5 * x_{11}^3 * x_{12}^2 * x_{13} - x_9 * x_{10} * x_{15} * x_{16}^2, x_0 * x_4 * x_{11}^3 * x_{12}^2 * x_{13} - x_8 * x_{10} * x_{15} * x_{16}^2,$
$x_0 * x_3 * x_{11}^3 * x_{12}^2 * x_{13} - x_7 * x_{10} * x_{15} * x_{16}^2, x_0 * x_2 * x_{11}^3 * x_{12}^2 * x_{13} - x_6 * x_{10} * x_{15} * x_{16}^2,$
$x_0 * x_1 * x_{11}^3 * x_{12}^2 * x_{13} - x_5 * x_{10} * x_{15} * x_{16}^2, x_0^2 * x_{11}^3 * x_{12}^2 * x_{13} - x_4 * x_{10} * x_{15} * x_{16}^2, x_2 * x_{10} * x_{13} * x_{14}^2 * x_{15}^3 * x_{16}^4 - x_0^2 * x_{11},$
$x_1 * x_{10} * x_{12} * x_{13}^2 * x_{14}^3 * x_{15}^4 * x_{16}^5 - x_0^2, x_{10} * x_{11} * x_{12}^2 * x_{13}^3 * x_{14}^4 * x_{15}^5 * x_{16}^6 - x_0)$,

在未来的工作中，我们将研究如何通过优化 Gr¨öbner 基的计算算法（例如 Faug`ere F_4 [9] 和 F_5 [10]算法）来改善这种情况。

5.2. 随机网络设计。 我们还在一个小型随机网络设计问题上测试了我们的方法，该问题表述如下：

$$\min \sum_{a\in A} c_a x_a + \frac{1}{N}\sum_{i=1}^{N}\sum_{c\in C}\sum_{a\in A} q_{ac} y_{ac}^i \tag{5.4}$$

$$\text{s.t.} \sum_{\substack{a\in A\\ a(0)=v}} y_{ac}^i - \sum_{\substack{a\in A\\ a(1)=v}} y_{ac}^i = d_{v,c}^i \quad \forall (v,c,i) \in V\times C\times\{1,\dots,N\} \tag{5.5}$$

$$\sum_{c\in C} y_{ac}^i \leqslant u_a x_a \quad \forall (a,i)\in A\times\{1,\dots,N\} \tag{5.6}$$

$$x_a \in \{0,1\}, y_{ac}^i \in \mathbb{Z}_{\geqslant 0} \tag{5.7}$$

按照第5.1节中的相同步骤，我们将这两种算法与商业求解器进行比较。我们的实例涉及一个小型问题，包括三个顶点和三条弧。图5.3展示了采用每种方法计算机会成本矩阵所需的时间。当

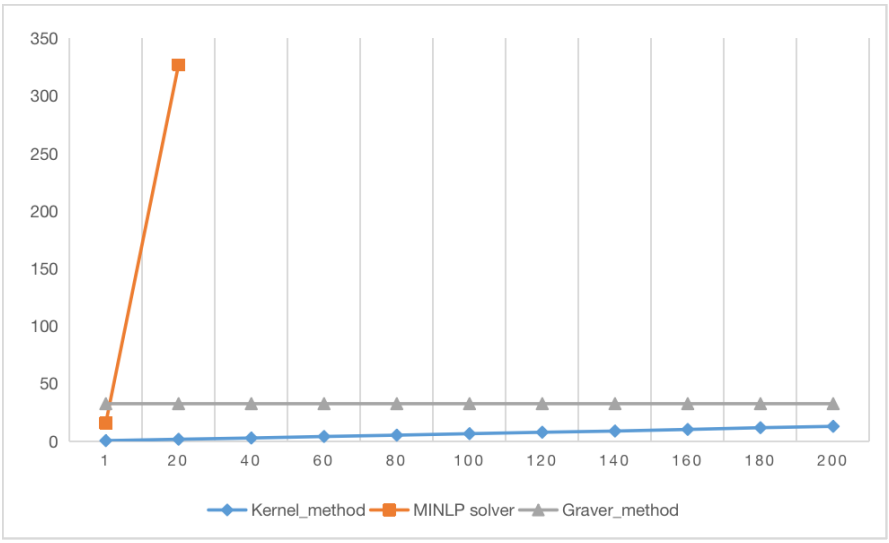


FIGURE 5.3. 子问题固定时计算成本的趋势（随机网络设计问题）

决策变量数量较少时。

6. CONCLUSION

本文介绍了使用代数方法计算 [15] 的机会成本矩阵的新算法，包括使用 Gr¨obner 基和 Graver 基。我们还给出了数学结果，展示了确定性和随机整数规划中 Graver 基之间的关系。数值结果表明，当基础整数规划的变量数量增长不太快时，随着情景数量增加，Gr¨obner 基的效率非常高。另一方面，当变量数量较多时，Graver 基方法很有意义。

在未来的工作中，我们计划纳入更新的 Gr¨obner 基计算方法，例如 Faug`ere 算法。我们还将研究如何把代数方法与拉格朗日割等近似方法相结合，从而进一步增加该方法能够处理的决策变量数量。

References

[1] Dimitris Bertsimas 和 Nishanth Mundru。“数据驱动的两阶段随机优化中的基于优化的情景约简”。《运筹学》（2022）。 ISSN：0030-364X。 DOI：10.1287/opre.2022.2265。[2] Dimitris Bertsimas、Georgia Perakis 和 Sridhar Tayur。“一种用于整数规划的新代数几何算法”。《管理科学》46（7，2000），第999–1008页。 ISSN：0025-1909。 DOI：10.1287/mnsc.46.7.999.12033。[3] Fausto Di Biase 和 R¨udiger Urbanke。“计算某些多项式环同态的核的算法”。《实验数学》4（3，1995），第227–234页。ISSN：1058-6458。 DOI：10.1080/10586458.1995.10504323。[4] John R. Birge 和 Fran¸cois Louveaux。《随机规划导论》。Springer New York，2011年。 ISBN：978-1-4614-0236-7。 DOI：10.1007/978-1-4614-0237-4。[5] Bruno Buchberger。《Gr¨obner基：多项式理想理论中的算法方法》。N K Bose 编。1985年。 DOI：10.1007/978-94-017-0275-1_4。[6] Pasqualina Conti 和 Carlo Traverso。Buchberger算法与整数规划。H. F. Mattson、T. Mora 和 T. R N Rao 编。1991年。 DOI：10.1007/3-540-54522-0_102。[7] David A. Cox、John Little 和 Donal O’Shea。《理想、簇与算法》。Springer International Publishing，2015年。 ISBN：978-3-319-16720-6。 DOI：10.1007/978-3-319- 16721-3。[8] Jitka Dupacov´a、Nicole Gr¨owe-Kuska 和 Werner R¨omisch。“随机规划中的情景约简”。Mathematical Programming 95（3，2003），第493–511页。 ISSN：0025-5610。 DOI：10.1007/s10107-002-0331-0。[9] Jean-Charles Faug`ere。“一种计算Gr¨obner基的新型高效算法（F4）”。Journal of Pure and Applied Algebra 139（1-3，1999），第61–88页。 ISSN：00224049。 DOI：10.1016/S0022-4049(99)00005-5。[10] Jean-Charles Faug`ere。“一种无需化约为零即可计算Gr¨obner基的新型高效算法（F5）”。ACM，2002年，第75–83页。 ISBN：1581134843。 DOI：10 . 1145 / 780506.780516。[11] Jack E. Graver。“线性规划与整数线性规划基础论 I”。Mathematical Programming 9（1，1975），第207–226页。 ISSN：0025-5610。 DOI：10.1007/ BF01681344。[12] Raymond Hemmecke。“关于正和性质及Graver测试集的计算”。Mathematical Programming 96（2，2003），第247–269页。 ISSN：0025-5610。 DOI：10.1007/s10107-003-0385-7。

[13] Raymond Hemmecke 和 R¨udiger Schultz。“随机整数规划中测试集的分解”。Mathematical Programming 94（2-3，2003），第323–341页。 ISSN：0025-5610。 DOI：10.1007/s10107-002-0322-1。[14] Ren´e Henrion、Christian K¨uchler 和 Werner R¨omisch。“基于差异距离的随机规划情景约简”。Computational Optimization and Applications 43（1，2009），第67–93页。 ISSN：0926-6003。 DOI：10.1007/s10589-007- 9123-z。[15] Mike Hewitt、Janosch Ortmann 和 Walter 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（2，2022），第747–771页。 ISSN：15729338。 DOI：10.1007/s10479-020-03843-x。[16] Julien Keutchayan、Janosch Ortmann 和 Walter Rei。“随机优化中的问题驱动情景聚类”（2021）。[17] Anton J. Kleywegt、Alexander Shapiro 和 Tito Homem de Mello。“随机离散优化的样本平均近似方法”。SIAM Journal on Optimization 12（2，2002），第479–502页。 ISSN：1052-6234。 DOI：10.1137/S1052623499363220
.
[18] Werner R¨omisch。《随机规划中的情景约简技术》。由 Osamu Watanabe 和 T Zeugmann 编。2009 年。 DOI：10.1007/978-3-642-04944-6_1.[19] R¨udiger Schultz。“具有随机技术矩阵和完全整数补偿的随机规划中的结构与稳定性”。《数学规划》70（1995），第 73–89 页。[20] R¨udiger Schultz、Leen Stougie 和 Maarten H. van der Vlerk。“通过枚举求解具有整数补偿的随机规划：一种使用格罗布纳基的方法”。《数学规划》83（1–3，1998），第 229–252 页。 ISSN：0025-5610。 DOI：10 .1007 / BF02680560。[21] Bernd Sturmfels。“整数规划的代数方法”。2004 年，第 99–113 页。[22] Bernd Sturmfels。《格罗布纳基与凸多面体》。美国数学学会，1996 年。[23] Rekha R. Thomas。《整数规划中的格罗布纳基》。1998 年。 DOI：10.1007/978-1-4613-0303-9_8.[24] R.R. Thomas 和 R. Weismantel。“整数规划中的截断格罗布纳基”。《工程、通信与计算中的应用代数》8（4，1997），第 241–256 页。 DOI：10.1007/s002000050062。

Yuchen Ge, Shandong University, Jinan, China
Email address: gycdwwd@163.com*URL*: https://gycdwwd.github.io/

Janosch Ortmann, CRM, GERAD and Université du Québec à Montréal, Case postale 8888, succ. Centre-ville Canada , Montréal (QC) H3C 3P8,*Email address*: ortmann.janosch@uqam.ca*URL*: http://crm.umontreal.ca/˜ortmann/

Walter Rei, CIRRELT and Université du Québec à Montréal, Case postale 8888, succ. Centre-ville, Montréal (QC) H3C 3P8, Canada*Email address*: rei.walter@uqam.ca