多项式对称性破除约束的自动生成

Mădălina Eraşcumadalina.erascu@e-uvt.ro 信息学院，蒂米什瓦拉西大学 蒂米什瓦拉，罗马尼亚

Johannes M iddekejoha nnes.middeke@tuj.temple.edu 天普大学日本校区 东京，日本

arXiv:2602.08297v1 [cs.SC] 9 Feb 2026

摘要

整数规划中的对称性会导致冗余搜索，通常借助对称性破除约束来处理，以尽可能多地剔除等价解。我们提出一种代数方法，可生成一族随机多项式不等式，用作对称破缺器。该方法以任意基多项式以及针对该整数规划问题的置换群作为输入。相关计算可在任何主流符号计算软件中轻松完成。为检验该方法，我们针对接近半容量、具有显著对称性的0-1装箱实例开展案例研究。我们静态生成随机二次对称性破缺约束，并将其加入基准整数规划问题，随后用Gurobi求解。结果表明，简单的对称破缺器，尤其是仅组合少量变量与排列的约束，最能稳定地缩短求解时间。

CCS 概念：• **计算方法** → **符号与代数算法**；**优化算法**；• **计算数学** →求解器。

关键词：代数对称破缺、整数规划、置换群、装箱问题

ACM 参考格式：
Mădălina Eraşcu 和 Johannes Middeke。2026。《多项式对称性破缺约束的自动生成》。载于《论文集》。 ACM，纽约，纽约州，美国， 10页。 https://doi.org/XXXXXXX.XXXXXXX

允许为个人或课堂使用而免费制作本作品全部或部分内容的数字副本或纸质副本，但不得出于营利或商业目的制作或分发副本，且副本首页必须保留本声明和完整引文。本作品中由作者以外的其他人拥有版权的部分，其版权必须得到尊重。注明出处后可以摘录。若要以其他方式复制、重新出版、发布到服务器或重新分发至列表，须事先获得明确许可和/或支付费用。请通过 permissions@acm.org 申请许可。，© 2026 版权所有者/作者保留版权。出版权授权给 ACM。
ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MMhttps://doi.org/XXXXXXX.XXXXXXX

1 引言

符号计算利用问题的对称性来求解问题，而在 [13], 优化和自动推理（约束规划 [8, 26], 混合整数规划 [14, 18, 25],可满足性 [1, 3],和可满足性模理论 [4]）中，通常将相同的对称性视为搜索树中的冗余分支，并通过显式的对称性破除约束将其剪除。为解决这一问题，一种常见且有效的对称性处理方法是添加约束，尽可能多地消除对称解。这些约束即所谓的对称性破除约束，可采用静态方式 [4, 24] 或动态方式 [2, 7, 9]。

我们的工作关注静态对称性破除，即先验地 添加约束，将可行解限制为各轨道的代表元。这是一个成熟的研究领域：在约束规划（CP）中，字典序引领者及相关的排序约束已是标准方法 [8,26]。在混合整数规划（MIP）中，结构最清晰的静态方法是多面体方法：对于行或列可互换的指派矩阵形式，轨道多面体刻画了按字典序排序的 0-1 矩阵的凸包，并给出强有力的线性描述和紧凑的扩展形式 [6, 14]；更一般地，也可以利用计算群论（稳定子链、Schreier-Sims 表）推导静态不等式，从而在任意形式中破除对称性 [16, 25]。在可满足性模理论（SMT）中，对称性破除通常施加于布尔抽象上 [4]。

尽管现有关于对称性破除的研究主要集中在线性不等式上，但在这项工作中，我们探索非线性对称破除器在计算上的有效性。我们从问题变量的一个基多项式出发，利用问题的对称群构造不等式。具体而言，给定 $h(x)$（一个多项式）以及 P（对称群中的一个变量置换），我们证明不等式 $h(Px) - h(x) \leq 0$ 可以作为静态对称性破除器（见定理 2.3）。请注意，在特殊情况下，若 $h(x)$ 是线性多项式，我们的方法将生成与现有方法（例如 [19]）所得结果相似的对称破除器。然而，我们的方法并不限于线性情形。

作为一个具体案例研究，我们将多项式对称性破缺方法应用于经典 0-1 装箱模型，并重点研究物品大小

集中在 $B/2$附近，其中 B 为箱大小。这一问题的动机来自虚拟机到服务器分配中的每箱两个作业（TBP2）时序装箱。由于存在许多等价的配对方式和箱标签排列，因而具有较高的对称性 [22]。

为了评估所提出的对称性破缺方法，我们生成了合成的近半容量装箱问题基准测试系列。随后，我们通过从一组选定的模板中生成一个基多项式，并对基准测试的对称群中的变量随机应用一组置换，为每个基准测试创建一系列随机多项式对称性破缺约束。

实验使用 Gurobi 12.0.3 [11] 在搭载 Apple M2 Pro 的系统上进行，并在 1800 个工作单位的限额下，通过 Gurobi 工作单位（确定性计算工作量）衡量性能。结果表明，二次对称性破除约束始终优于线性对称性破除约束，而且重要的是，也胜过 Gurobi 内置对称性破除。最可靠的性能提升来自小规模二次破坏器集合（变量和置换数量较少），其中 xy 模板的整体表现最佳。相比之下，纯线性对称性破除约束往往会增加求解器工作量，而较大的破坏器集合由于额外非线性约束带来的开销，表现出更大的波动。

本文的贡献如下：

- 我们提出一种对称性破除方法，针对群置换 P和选定的基多项式 h，生成形如 $h(Px) - h(x) \leq 0$ 的约束。与基于线性排序或多面体约束的标准 CP/MIP 方法不同，我们的构造自然地产生非线性对称性破缺约束。
- 我们的方法通过比较群置换下的多项式求值来破除对称性；所得的二次模板引入非线性相互作用，而不是将现有方法重新编码到符号计算框架中。因此，我们的方法并非增量式的，而是新颖的。
- 我们的方法是首个自动生成非线性对称性破缺约束的方法——前提是已知对称群（或其子群）。此外，实验结果提供了二次对称性破缺约束高效的经验证据。
- 我们的方法对于对称破缺器并非先验地已知的问题尤其有价值[1]，从而推动未来在装箱问题之外进行评估。

本文结构如下。 第 2 节 介绍整数规划的主要对称性破缺结果。第 3 节将其应用于装箱模型（基准测试、对称子群和对称性破除器的生成）。 第 4 节 报告实验评估结果。 第 5 节 总结全文并展望未来工作。

[1]众所周知，装箱问题具有强有力、经过充分研究且计算上有效的线性对称性破除约束（例如，基于字典序的代表元）。

2 主要结果

我们考虑如下形式的整数规划问题

$$\begin{aligned} \text{minimise} \quad & f(x) \\ \text{subject to} \quad & Ax \geq b, \\ & x \in U^n \end{aligned} \tag{IP}$$

其中，$f\colon\ \mathbb{R}^n \to \mathbb{R}$ 是一个多项式函数，$A \in \mathbb{R}^{m\times n}$ 是一个矩阵，$b \in \mathbb{R}^m$ 是一个向量，x是决策变量向量，$U \subset \mathbb{R}$ 是一个有限集。参见例如 [19]。

整数规划通常使用树搜索方法求解，例如分支定界法。这类方法通过对变量赋值进行分支并利用界剪除子问题，从而探索搜索树。如果模型具有对称性，那么许多分支就对应于彼此置换后等价的解，因此求解器可能会反复探索实质上相同的子树。这种冗余会大幅增加搜索工作量，因此需要采用对称性破除约束，将求解器限制为每个对称轨道只搜索一个代表。

由于文献中对对称性的定义存在许多细微差异（例如，参见 [8] 以了解综述），因此有必要明确本文所采用定义的确切含义。与问题描述（IP）一样，该定义基于 [19]中给出的定义。

定义 2.1 (对称性)。 若 $n\times n$ 置换矩阵P 满足对称性 （IP）的条件是 $f(Px) = f(x)$ 且$A(Px) \geq b \iff Ax \geq b$ 对所有 $x \in U^n$ 均成立。给定问题的所有对称性在乘法运算下构成一个群，我们称之为（*IP*）的对称群。

我们将（IP）的对称群记为G。此外，我们将 Π_n 记为所有 $n\times n$ 置换矩阵的集合。

如果 $x^* \in U^n$ 是（IP）的一个最优解，且 $P \in G$ 是（IP）的一个对称性，那么 Px^* 也是一个最优解。这会导致搜索中出现冗余。因此，人们通常会尝试避免生成那些通过对称性与其他解相关联的解。

定义 2.2 （对称性破除器）。 对决策变量中的一个谓词 ψ: $U^n \to \{$真、假$\}$，若存在一个最优解 $x^* \in U^n$ 和一个对称性 $P \in G$，使得 $\psi(x^*)$ = 为真 而 $\psi(Px^*)$ = 为假，则称其为对称性破除器。

在静态对称性破除中，对称性破除器会先验地添加到问题（IP）中。更形式化地说，给定一个对称性破除器 ψ（x），我们得到一个新问题

$$\begin{aligned} \text{minimise} \quad & f(x) \\ \text{subject to} \quad & Ax \geq 0, x \in U^n, \psi(x). \end{aligned} \tag{IP'}$$

由于 ψ（x）对（IP）的至少一个最优解成立，因此求解（IP'）足以找到（IP）的一个解。

可以向问题中添加不止一个对称性破除器。但在这种情况下，必须注意这些对称破缺器彼此兼容，即我们必须确保至少存在一个最优解

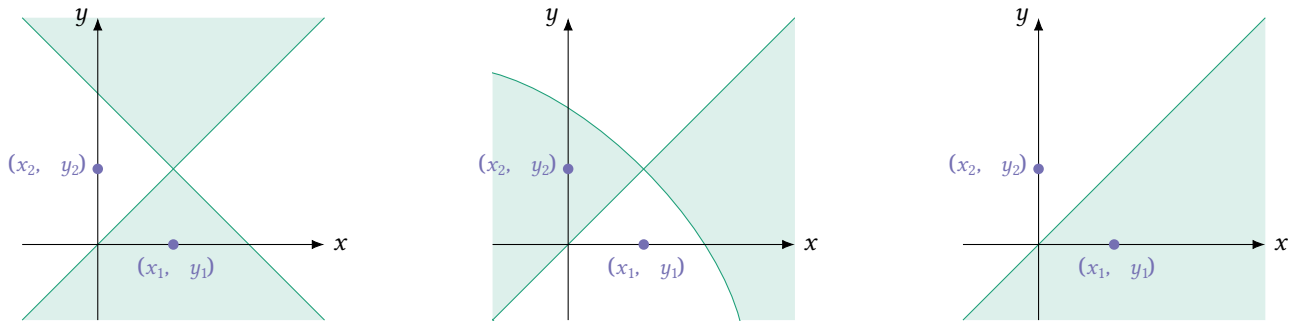

图 1. 区域 $2y + x^2 - 2x - y^2 \le 0$ (左侧)、 $y^3 - 3y + 3x - x^3 \le 0$ (中间)和 $y - x \le 0$ (右侧)以较深的色调显示。每种情况下，示例问题的一个最优解位于该区域内，而另一个则不在其中。

该最优解由所有添加的对称破缺器同时满足。否则，这些对称破缺器将使问题不可行。

虽然对称破缺器原则上可以采用任何形式，但本文将重点讨论多项式不等式。考虑多项式环 $R = \mathbb{R}[x]$ ，其中$x = (x_1, \dots, x_n)$ 是由 n 个变量组成的向量。对于多项式$h(x) \in R$ 以及置换 $P \in G$，我们将用 $h(Px)$表示通过依照 P 对 $h(x)$中的所有变量进行替换而得到的多项式。更确切地说，如果 P 将第 i 个单位向量 e_i 映射到第 j 个单位向量 e_j，那么我们就在 $h(Px)$中用 x_j 替换 x_i。

定理 2.3。 设 G 为 (IP) 的对称群，且设$R = \mathbb{R}[x]$ 其中 $x = (x_1, \dots\dots, x_n)$。那么，对于任意 $h(x) \in R$以及任意 $P \in G$，至少存在一个最优解 $y^* \in U^n$满足不等式

$$h(Px) - h(x) \le 0.$$

这意味着， $h(Px) - h(x) \le 0$ 可以作为对称性破除器添加到问题 (IP)中，而不会丧失找到最优解的能力。

证明。 假设 $x^* \in U^n$ 是（IP）的一个最优解。那么，轨道 $Gx^* = \{Px^* \mid P \in G\}$ 中的任意元素也都是最优解。此外， Gx^* 是有限的，因此集合$H = \{h(y^*) \mid y^* \in Gx^*\} \subseteq \mathbb{R}$也是有限的。因此， H 存在最大值。设$y^* \in Gx^*$满足 $h(y^*) =$最大 H。那么，我们有

$$\forall P \in G\colon h(Py^*) \le h(y^*).$$ □

注 2.4. 对于任意固定的多项式 $h \in R$， 定理 2.3允许我们使用任意多个排列 $P_1, \dots, P_\ell \in G$，从而生成完整的对称破缺器族

$$h(P_1x) - h(x) \le 0, \quad \dots, \quad h(P_\ell x) - h(x) \le 0.$$

此外，将同一族中的所有破除约束同时加入问题，也不会妨碍我们找到最优解，因为定理保证至少存在一个满足所有破除约束的解。

需要注意的是，基多项式 $h \in R$ 的选择对破除约束的效果有巨大影响。极端情况下，如果 h 在群 G作用下保持不变，那么破除约束就只是平凡的不等式$0 \le 0$。还需要注意的是， G 通常会非常大。因此，尽管理论上我们可以针对给定的任意基多项式计算所有可能的破除约束，但实际上这是不可行的。

例 2.5. 为说明定理 2.3，考虑这个二变量问题

$$\begin{aligned} \text{minimise} \quad & x + y \\ \text{subject to} \quad & x + y \ge 1, \\ & x, y \in \{0, 1\}. \end{aligned}$$

对于这个问题，置换群为G=$\{\mathbf{1}, M_{(1\,2)}\} \subseteq \Pi_2$其中 $M_{(1\,2)}$ 是对应于换位 $(1\ 2) \in S_2$的置换矩阵。我们随机选取基多项式$h = 2x + y^2 \in \mathbb{R}[x, y]$。由此得到对称性破除器

$$\begin{aligned} h(M_{(1\,2)}(x, y)^t) - h(x, y) &= h(y, x) - h(x, y) \\ &= 2y + x^2 - 2x - y^2 \le 0. \end{aligned}$$

该示例问题的两个最优解为 $(x_1, y_1) = (1, 0)$ 和 $(x_2, y_2) = (0, 1)$。我们可以验证， (x_1, y_1) 满足不等式 $2y + x^2 - 2x - y^2 \le 0$ ，而 (x_2, y_2) 不满足（见图 1）。

作为另一个基多项式，考虑 $h = x^3 - 3x$。在此情形下，不等式变为

$$y^3 - 3y + 3x - x^3 \le 0.$$

同样，我们可以验证，其中一个解，更确切地说是$(x_2,\ y_2) = (0, 1)$，满足这个新的不等式，而另一个则不满足（另见 图 1）。

最后，考虑线性基多项式 $h = 2x + y$。此处，不等式为

$$2y + x - 2x - y = y - x \le 0.$$

(再一次， 图 1 展示了不等式成立的区域。)

注 2.6。 如果我们从一个非零的线性基多项式开始 $h(x) = v_1x_1 + \ldots + v_nx_n$，那么我们的方法将生成线性对称性破除约束。事实上，不难证明，我们得到的线性对称性破除约束与向量 $\bar{x} = (v_1, \ldots, v_n)$ 在 [19,定理 173]中得到的完全相同。然而，我们的方法更具一般性，因为它并不局限于线性多项式。尤其是，如图 1 所示，如果允许使用非线性对称性破缺约束，我们就能得到种类丰富得多的几何形状。

在 [19] 中，为证明线性对称性破除约束的正确性而给出的证明， [10] 依赖于基本区域这一概念，该概念最初由此处引入。尽管我们 定理 2.3 的证明要简单得多，也不需要这些额外工具，但可以证明，我们的对称破缺器同样能够定义基本区域。为方便读者，我们在此重述定义。

定义 2.7 ([10, 第 3] 章)。 集合 $F \subseteq \mathbb{R}^n$ 相对于有限子群 $G \leq \mathrm{O}_n$ 是基本区域，当且仅当
1. F 是开集；2. 对所有 $P \in G \setminus \{\mathbf{1}_n\}$， $PF \cap \overline{F} = \emptyset$；3. $\bigcup_{P\in G} PF = \mathbb{R}^n$。

对于该定理，我们将讨论范围限制在由置换矩阵构成的群这一情形。不过，这一限制对于该结果而言并非必要。

定理 2.8. 设 $G \leq \Pi_n$ 是一个有限置换群。设 $h \in \mathbb{R}[x]$。对于每个 $P \in G$，定义

$$L_P = \{x \in \mathbb{R}^n \colon h(Px) - h(x) < 0\}.$$

如果对于每个 $P \in G$，当 $P \neq \mathbf{1}_n$ 时都有 $L_P \neq \emptyset$，那么 $F = \bigcap_{P\in G\setminus\{\mathbf{1}\}} L_P$ 是相对于 G 的基本区域。

证明的思路受到 [10,定理 3.1.2] 的启发。然而，尽管 [10]大量使用了线性代数概念，尤其是范数和正交映射，我们的证明并不局限于线性情形。

定理 *2.8* 的证明。我们首先注意到 L_P 是开集，因为 $x \mapsto h(Px) - h(x)$ 是连续函数。这意味着 F 是开集，因为它是有限个开集的交集。

注意， $F = \{x \in \mathbb{R}^n \colon h(Qx) < h(x) \ \forall Q \in G, \ Q \neq \mathbf{1}\}$。因此，我们有

$$\begin{aligned} PF = P(\{x \colon h(Qx) < h(x) \ \forall Q \neq \mathbf{1}\}) \\ = \{Px \colon h(Qx) < h(x) \ \forall Q \neq \mathbf{1}\} \\ = \{y = Px \colon h(QP^{-1}y) < h(P^{-1}y) \ \forall Q \neq \mathbf{1}\} \\ = \{y \colon h(Uy) < h(P^{-1}y) \ \forall U \neq P^{-1}\} \end{aligned}$$

因为当 Q 遍历 $G \setminus \{\mathbf{1}\}$ 中的排列时， $U = QP^{-1}$ 将遍历除 P^{-1} 之外的所有排列。因此，如果我们有 $x \in F \cap PF$，便会得到矛盾 $h(x) < h(P^{-1}x) < h(x)$，其中第一个不等式来自 $x \in PF$，其中 $U = \mathbf{1}$，第二个则来自 $x \in F$。因此，$F \cap PF = \emptyset$。

最后， $\overline{PF} = \{y \colon h(Uy) \leq h(P^{-1}y) \ \forall U \neq P^{-1}\}$对每个$P \in G$都成立。令 $x \in \mathbb{R}^n$ 为任意值。由于 G 是有限的，因此 $M = \{Px \colon P \in G\}$也是有限的。因此，存在一个 $P \in G$，使得 $h(P^{-1}x) = \max M$。于是 $x \in \overline{PF} \subseteq \bigcup_{Q\in G} QF$. □

注释 2.9。 根据 [10, 定理 3.1。2],总存在一个线性多项式 $h = a_1x_1 + \ldots + a_nx_n \in \mathbb{R}[x]$ 满足 $a_1, \ldots, a_n \in \mathbb{R}$，且符合条件 $L_P \neq \emptyset$ 对所有 $P \in G \setminus \{\mathbf{1}_n\}$在定理 2.8中。

3 案例研究

3.1 问题描述（装箱模型）

装箱问题 [20, 第 8] 章定义如下

$$\begin{aligned} \text{minimise} \quad & y_1 + \ldots + y_n \\ \text{subject to} \quad & \forall k = 1, \ldots, n \colon s_1x_{1k} + \ldots + s_mx_{mk} \leq By_k \\ & \forall i = 1, \ldots, m \colon x_{i1} + \ldots + x_{in} = 1 \\ & \forall i = 1, \ldots, m \ \forall k = 1, \ldots, n \colon x_{i,k} \in \{0, 1\} \\ & \forall k = 1, \ldots, n \colon y_k \in \{0, 1\} \end{aligned} \tag{BP}$$

其中 m， n为正整数，且 B， s_1，……， s_m为正实数。我们将 s_1，……， s_m称为（对象）的大小（BP），并将 B 称为箱大小。

3.2 基准生成（近半容量族）

受 [22]，启发，我们在接近半容量的实例静态装箱基准上评估了该方法，其中物品大小集中在 $B/2$ 附近（其中 B 为箱容量）。这一设置具有实际背景，源于每箱两个作业的时序装箱（TBP2），该问题出现在面向专用资源密集型云系统的虚拟机到服务器分配中。从计算角度看，这类实例颇具挑战性，因为它们存在大量等价的物品配对、任意的箱标签置换，而且几乎所有对象都无法区分。

我们为装箱问题（BP）构建了一组 B合成实例族，并针对每个实例族报告箱子容量 n、物品数量以及物品尺寸规格（此处采用接近容量一半的范围）。表 1总结了所选设置，使基准问题的运行时间约为5 分钟。生成的基准测试包含约 400 万个变量和约 4000 个约束。

表 1. 近半容量实验设置

类别	3	5	7	9
n	2000	2000	1024	1000
区间	[49,51]	[48,52]	[47,53]	[46,54]

3.3 对称群

对于对称破缺器，我们首先需要确定装箱问题的对称群（或其中一个足够大的子群）。

我们将用 $e_1 = (1, 0, \ldots\ldots, 0)$ $e_2 = (0, 1, 0, \ldots\ldots, 0)$……表示标准基向量（长度任意），用 $e = (1, \ldots\ldots, 1)$表示全一向量（长度同样任意）。此外，我们用 1_q 表示 $q \times q$ 单位矩阵。我们将 $\{1, \ldots\ldots, q\}$ 上的对称群记为 S_q。对于 $\tau \in S_q$，我们用 M_τ 表示 $q \times q$ 置换矩阵，其中$M_\tau e_{\tau(i)} = e_i$表示所有的 $i = 1, \ldots\ldots, q$。此外，对于 $\tau \in S_q$，我们定义 $\hat{} M_\tau = \mathrm{diag}(1, M_\tau) \in \mathrm{GL}_{q+1}(\mathbb{R})$。

将 (BP) 的变量汇集到一个$(m+1) \times n$矩阵中会很方便

$$V = \begin{pmatrix} y_1 & \cdots & y_n \\ \hline x_{11} & \cdots & x_{1n} \\ \vdots & & \vdots \\ x_{m1} & \cdots & x_{mn} \end{pmatrix}.$$

这使我们能够用克罗内克积和 vec 向量化函数来表示目标函数和各项条件。（例如，关于符号及其基本性质，参见 [12, 第 11.4] 节。）由此可将目标函数写为 $(e \otimes e_1)\,\mathrm{vec}(V)$。类似地，利用 $u = (B, -s_1, \ldots\ldots, -s_m)$，我们可以将约束写为 $(1_n \otimes u)\,\mathrm{vec}(V) \geq 0$ 和 $(e \otimes (0\ 1_m))\,\mathrm{vec}(V) = e$.（这里，$(0\ 1)$ 表示一个零列后接一个单位矩阵。）

在第 2 节的实验中，我们考虑两类置换矩阵。对于第一类，任取一个置换 $\tau \in S_n$。那么

$$\begin{aligned}(e \otimes e_1)(M_\tau \otimes 1_{m+1})\,\mathrm{vec}(V) &= (eM_\tau \otimes e_1)\,\mathrm{vec}(V) \\ &= (e \otimes e_1)\,\mathrm{vec}(V).\end{aligned}$$

因此，将置换矩阵 $M_\tau \otimes 1_{m+1}$ 作用于变量向量 $\mathrm{vec}\,V$不会改变目标函数。类似地，$(e \otimes (0\ 1_m))(M_\tau \otimes 1_{m+1}) = (e \otimes (0\ 1_m))$，这意味着 $(M_\tau \otimes 1_{m+1})$ 也使第二个条件保持不变。最后，

$$\begin{aligned}(1_n \otimes u)(M_\tau \otimes 1_{m+1})\,\mathrm{vec}(V) \\ = M_\tau (1_n \otimes u)\,\mathrm{vec}(V) \geq M_\tau 0 = 0\end{aligned}$$

由于 M_τ 只改变了不等式的顺序，因此这等价于第一个条件。由此可得，矩阵 $M_\tau \otimes 1_{m+1}$ 属于（BP）的对称群。

对于第二类排列，我们作如下假设：尺寸已按升序排列；即 $s_1 \leq \ldots \leq s_m$。考虑下标 $1 = i_1 < \ldots < i_{\ell+1} = m+1$，使得对所有 $j = 0, \ldots, \ell$有$s_{i_j} = \ldots = s_{i_{j+1}-1}$，并且对 $j = 2, \ldots, \ell$有$s_{i_j - 1} < s_{i_j}$。我们将 $i_1, \ldots, i_{\ell+1}$ 称为给定装箱问题实例的尺寸边界。我们将$S_{i_1, \ldots, i_{\ell+1}}$ 记为排列 $\sigma \in S_m$ 的子群，使得对所有 $j = 1, \ldots, \ell$，有 $\sigma(\{i_j, \ldots, i_{j+1}-1\}) = \{i_j, \ldots, i_{j+1}-1\}$。（也就是说，对任意$\sigma \in S_{i_1, \ldots, i_{\ell+1}}$以及任意 $j = 1, \ldots, m$，我们有$s_j = s_{\sigma(j)}$。）对于这样的$\sigma \in S_{i_1, \ldots, i_{\ell+1}}$，考虑置换矩阵$(1_n \otimes \hat{} M_\sigma)$。我们有$(e \otimes e_1)(1_n \otimes \hat{} M_\sigma) = (e \otimes e_1)$，这表明$(1_n \otimes \hat{} M_\sigma)$使目标函数保持不变。对于第一个条件，我们得到$(1_n \otimes u)(1_n \otimes \hat{} M_\sigma) = (1_n \otimes u)\hat{}$，因为根据对所有$j = 1, \ldots, m$均有$s_j = s_{\sigma(j)}$这一假设，$uM_\sigma = u$。最后，$(e \otimes (0\ 1_m))(1_n \otimes \hat{} M_\sigma) = M_\sigma (e \otimes (0\ 1_m))$，这与 $M_\sigma e = e$ 一起表明第二个ˆ条件在$(1_n \otimes M_\sigma)$下同样保持不变。因此，这类矩阵也是(BP)的对称群的成员。

3.4 对称破除器生成

对于表 1中的近半容量实例族，我们通过实例化一组模板（见表 2），并通过四个控制变量数量和抽样置换数量的子类调整模板规模，来生成线性和二次对称性破除约束。对于这些模板，我们使用一种简写记号，其中每个 x 表示一个随机线性多项式，包含$x_{i,\ k}$、 i、 $k = 1$、…… n；而每个 y 表示一个关于 y_1、……、 y_n的随机线性多项式。这些随机多项式的系数限定为 0 和 1。此外，模板中的任何x^2 或 y^2 均指两个不同线性多项式的乘积。

表 2. 实验中使用的对称破缺约束模板及其规模。

类别	模板
线性	$x, y, x+y$
二次	x^2, y^2, xy, x^2+y^2
混合	$x+y^2,\ \ x^2+y$

Size	#变量	#排列
变量较少，排列较少	≈ 10	50
变量较少，排列较多	≈ 10	500
变量多，置换少	≈ 1000	50
变量众多，置换少	≈ 4000	50

请注意，不同模板中的变量数量略有不同。例如，对于 x^2、 y^2 和 xy 模板，“变量少”的情况使用 9 个变量（即，我们构造由 3 个变量组成的线性多项式的乘积）。对于 $x^2 + y^2$模板，我们使用 18 个变量； $x + y^2$ 和 $x^2 + y$ 使用 16 个变量；而 x、 y 和 $x+y$使用 10 个变量。同样，“变量多”和“变量众多”的情况下，变量的确切数量也有所不同。

对于每种形状和每种尺寸，我们生成一个随机基多项式 h，然后从对称群中选取 50（或 500）个置换P，以获得对称性破除约束 $h(Px) - h(x) \leq 0$。对于 $x^2 + y$ 和 $x + y^2$ 形状，我们只保留包含二次项的破除约束。（如果不进行此筛选，二次项可能会相互抵消，我们

最终可能只得到一组线性多项式。然而，我们希望单独测试线性对称性破除约束。）

类型为 $M_\tau \otimes \mathbf{1}_{m+1}$ 且带有 $\hat{}\tau \in S_n$ 的排列有 $n!$ 个。对于类型为 $\mathbf{1}_n \otimes M_\sigma$ 且带有$\sigma \in S_{i_1,\dots,i_{\ell+1}}$ 的排列，由于 $S_{i_1,\dots,i_{\ell+1}} \cong \bigtimes_{j=1}^{\ell} S_{i_{j+1}-i_j}$，有$\prod_{j=1}^{\ell}(i_{j+1}-i_j)$ 种选择。由于这两类排列彼此可交换，因此在 G 中总共有 $n!\ \prod_{j=1}^{\ell}(i_{j+1}-i_j)$ 个排列。在我们的实验中，这个数大得不切实际。例如，对于包含 3 个类别、 $m = n = 2000$ 且尺寸边界为 $0, 688, 1320, 2000$ 的例子，排列数量超过 $10^{10\,520}$。

因此，为了生成排列，我们将采用不同的策略。对称群由所有形如 $M_{(1\,k)} \otimes \mathbf{1}_{m+1}$ 且带有 $\hat{}k = 2,\dots,n$ 的（对换）矩阵，以及所有形如 $\mathbf{1}_n \otimes M_{(i_j\,k)}$ 的矩阵生成，其中 $j = 1,\dots,\ell$ 且 $k = i_j+1,\dots,i_{j+1}-1$。对于每个基多项式$h$，我们只需随机选取其中 50 个生成元（允许重复），并将它们相乘。然后将所得结果作用于基多项式 h。

4 实验结果

4.1 实验设置

我们在近半容量装箱问题族上评估了框架生成对称破缺器的效果（见表 1）。所有实验均使用 Gurobi Optimizer 12.0 运行。3 [11]（学术许可证），运行平台为 Apple M2 Pro（10 核）。表 3 总结了实验中评估的四种 Gurobi 配置。所有运行均启用了非凸二次处理。基准配置同时禁用了预处理和对称性处理，作为参考点[2]。 默认配置在预处理和对称性处理方面均采用 Gurobi 自动设置。为研究 Gurobi 中对称性破缺强度的影响，我们将预处理保持为自动设置，并调整对称性：保守采用温和的对称性策略，而激进则采用所考虑的最强对称性处理。请注意

表 3. 实验中使用的 Gurobi 配置

参数 基准		默认	保守	激进
非凸	2	2	2	2
预处理	0	-1	-1	-1
对称性	0	-1	1	2

即使禁用了上述两个选项，Gurobi 仍启用了一系列选项（例如割平面、启发式），这些选项会在树中应用搜索方法之前使用。

我们报告 Gurobi 工作单位（截止时间为 1800），这是一项内置统计指标，用于量化计算工作量（这些工作量）花在优化上的程度，并提供一种确定性的替代方案，以取代墙钟运行时间[3]。

对于非线性对称性破缺约束，由于 SMT 求解器不支持非线性约束，因此不适用于这种情形。因此，我们在这项初步研究中专注于混合整数规划求解。能够处理非线性约束的约束规划方法（例如 CPLEX [17]）留待未来研究。

基准实例、生成的对称破缺器，以及用于复现实验并独立验证实验结果的脚本，均已在我们的 GitHub 仓库 [5] 中公开。

4.2 实验中使用的对称性破除器构造

我们使用表 2 中的模板实例化对称性破除器生成方法，并通过四个子类来控制变量数量和抽样置换数量，从而改变规模。

生成对称破缺器时，我们最初使用了 Python [21] 中的 SymPy 库。然而，对于实验所用的输入规模和不同测试用例的数量，Python 代码运行速度太慢。因此，我们改用编译型通用编程语言 OCaml [15] 实现该方法。给定装箱问题实例的尺寸边界，我们的自定义程序可在一至三分钟内生成数百个对称破缺器族。实现代码可在 GitHub 仓库 [5] 中找到。

4.3 结果解读

我们总共评估了 1440 个问题实例，这些实例由多个基准类别、破坏器模板和规模组合而成，且均基于基准配置。总体而言，超时次数很少：只有 10 次运行（约 0.7%）达到了时间上限。其中绝大多数发生在线性对称性破除约束的设置中，而二次配置中仅观察到一次超时。

图 2 汇总了所有实例类别的实验结果，并概述了每种对称性破坏器模板和尺度下 Gurobi 的相对工作单位。所有数值均以未添加额外对称性破除约束的基线设置为基准进行归一化，该设置以 100% 处的紫色实线表示。我们还报告了 Gurobi 默认设置在各类别中的相对耗时，并以绿色虚线表示。保守和激进配置未能在分配的工作单位内完成，因此未显示在图中。

[2]根据在线文档，预处理会简化模型。内置的对称性破除方法可能会干扰我们的对称破缺器。

[3]Gurobi 将工作单位定义为求解器所耗计算工作量的确定性度量，旨在使其比运行时间更不易受到机器负载和硬件差异的影响。

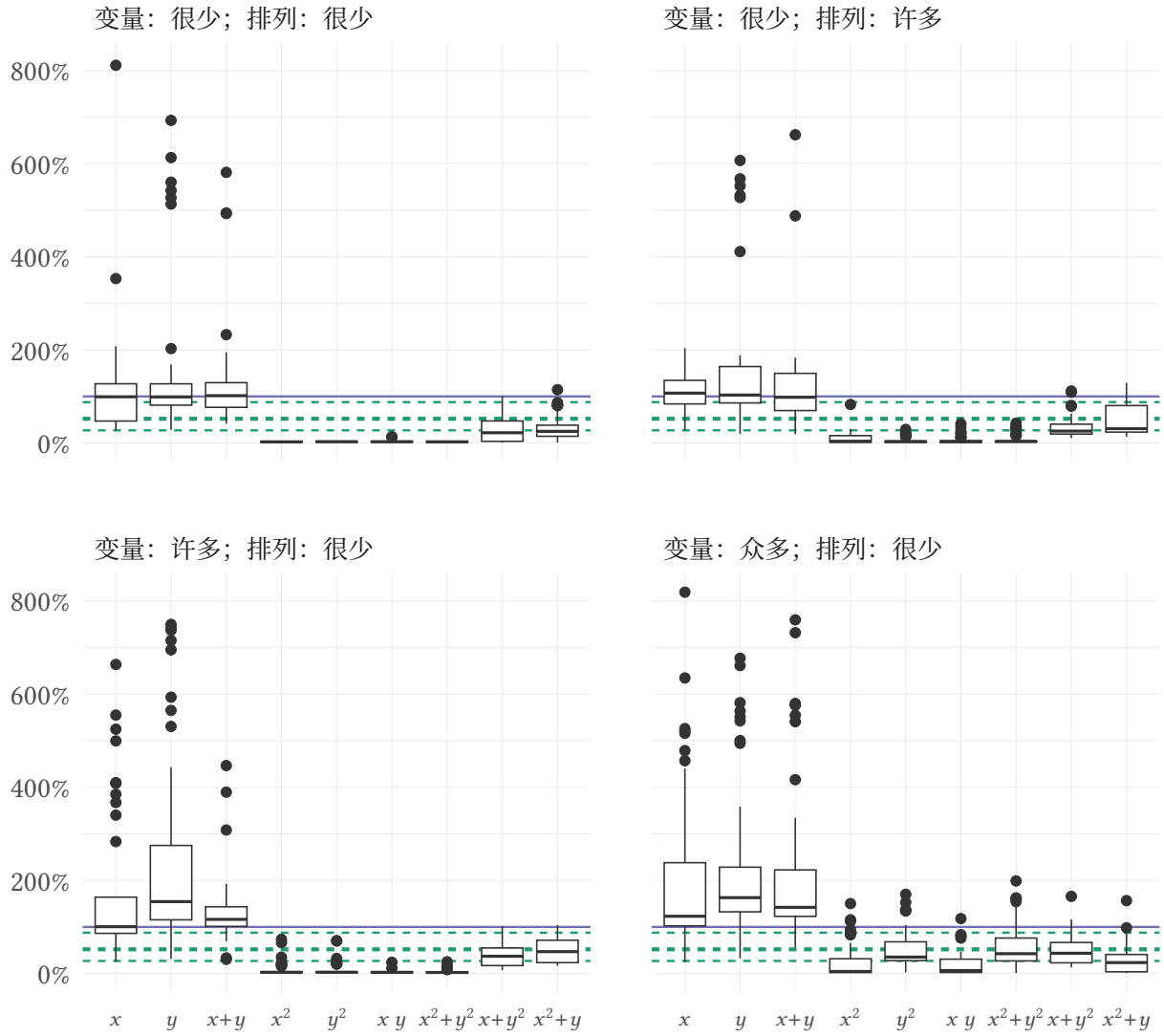

图 2. 所有测试用例（3、5、7 或 9 个类别）的综合结果。该图总结了每种破坏器形状以及每种经过测试的变量数量和排列数量组合对应的工作单位。工作单位以基准问题的工作单位为参照进行缩放（即，不含任何破坏器且类别数量相同的版本）。基准以紫色实线表示，Gurobi 对每个测试用例采用默认设置时的相对耗时以绿色虚线表示。

几个明显的趋势由此显现。首先，二次对称性破缺约束的表现始终优于线性对称性破除约束。此外，它们的表现也超过了 Gurobi 内置的对称性处理。具体而言，小尺度（即变量较少、排列较少）的 xy 模板优于线性对称性破除约束和 Gurobi 内置的对称性处理。

其次，小尺度对称性破坏器集合（包括线性和二次集合）在所有模板中都是最可靠且始终有益的。

第三，纯线性对称性破除约束往往效率较低，而且常常会增加求解器工作量，尤其是在变量数量或置换数量增加时。这一现象在所有四种规模设置下都一致。

第四，随着对称破缺器规模增大，无论是涉及更多变量，还是采样更多排列，性能都会变得更加不稳定。虽然一些二次模板仍能带来改进，但较大的破坏器集合也可能降低求解器的速度，这表明对称性约简与额外非线性约束带来的开销之间存在权衡。这种影响在“many variables”和“numerous variables”设置下最为明显。

5 结论与未来工作

我们提出了一种静态对称性破缺方法，可自动生成约束。作为输入，我们只需要

,
,

知道问题的对称群（或其一个子群）。随后，我们使用固定模板，在决策变量中生成一个随机多项式 $h(x)$，并应用来自对称群的随机置换 P，以寻找一族形式为$h(Px) - h(x) \leq 0$的不等式。由于该方法基于简单的代数运算，因此可以轻松地使用任何符号计算软件实现。与现有方法不同，我们的技术自然地产生非线性对称性破缺约束，并引入非线性交互，而非对现有线性方案重新编码。在近半容量装箱问题中，静态生成的二次对称性破缺约束始终优于基准；规模最小的破坏器最为可靠，混合双线性交叉项模板效率最高，而规模较大的破坏器集合表现更不稳定，有时还会拖慢求解器。

作为未来工作，我们计划开展以下工作。(1) 改进置换生成算法。目前，我们采用的置换完全是随机的，而且通常不会改变多项式 $h(x)$。这意味着许多生成的不等式都是平凡的，需要将其舍弃。尝试五十次，有时找到的非平凡不等式还不到十个。我们可以限制随机生成的范围，只考虑不会使出现在 $h(x)$中的变量保持不变的置换，从而改进这一点。(2) 评估该方法在三次对称性破除器上的效果。这是一项编程工作，因为我们需要以表达式树的格式生成对称性破除器 [23]，Gurobi 正是以这种方式表示三次约束。(3) 评估我们的方法在任意函数（例如代数函数或超越函数）上的效果。(4) 评估 IBM ILOG CPLEX Optimization Studio 等约束规划求解器 [17]。(5) 我们还计划将对称性破除生成方法扩展到装箱物品彼此不独立的场景。这类情况常见于在云中部署基于组件的应用程序和微服务系统时，部署过程需要满足组件间约束。

参考文献

[1] Fadi A. Aloul、Igor L. Markov 和 Karem A. Sakallah。2003 年。Shatter：用于布尔可满足性的高效对称性破缺。载于 第 *40* 届设计自动化会议（*DAC*）论文集。ACM，836–839。 doi：10.1145/775832.776028[2] Rolf Backofen 和 Sebastian Will。1999 年。在基于约束的搜索中排除对称性。载于 约束规划原理与实践国际会议。Springer，73–87。 [3] James M. Crawford、Matthew L. Ginsberg、Eugene M. Luks 和 Amitabha Roy。1996 年。用于搜索问题的对称性破缺谓词。载于 第五届知识表示与推理原理国际会议论文集（*KR'96*）。Morgan Kaufmann，148–159。[4] Saket Dingliwal、Ronak Agarwal、Happy Mittal 和 Parag Singla。2019 年。SAT 模理论对称性破缺的进展。 *arXiv* 预印本 *arXiv:1908.00860*（2019）。 https://arxiv.org/abs/1908.00860

[5] Mădălina Eraşcu 和 Johannes Middeke。2026。多项式对称性破缺器。https://github.com/merascu/PolynomialSymmetryBreakers [6] Yuri Faenza 和 Volker Kaibel。2009。用于装箱和划分轨道多面体的扩展表述。《运筹学数学》34，3（2009），686–697。[7] Torsten Fahle、Stefan Schamberger 和 Meinolf Sellmann。2001。对称性破缺。载于《约束规划原理与实践国际会议》。Springer，93–107。[8] Ian P. Gent、Karen E. Petrie 和 Jean-François Puget。2006。约束规划中的对称性。载于《约束规划手册》，Francesca Rossi、Peter van Beek 和 Toby Walsh（编）。爱思唯尔，329–376。[9] Ian P. Gent 和 Barbara M. Smith。2000。约束规划中的对称性破缺。载于《第14届欧洲人工智能会议论文集》（德国柏林）（ECAI'00）。IOS出版社，NLD，599–603。[10] Larry C. Grove 和 Clark T. Benson。1985。《有限反射群》（第2版）。《数学研究生教材》第99卷。Springer Science+商业媒体。doi:10.1007/978-1-4757-1869-0[11] Gurobi Optimization, LLC。2024。Gurobi 优化器参考手册。https://www.gurobi.com [12] Leslie Hogben（编）。2014。《线性代数手册》（第2版）。CRC出版社，博卡拉顿。doi:10.1201/b16113[13] Evelyne Hubert。2024。在代数计算中保留并利用对称性。载于《2024年符号与代数计算国际研讨会论文集》。13–13。[14] Volker Kaibel 和 Marc Pfetsch。2008。装箱与划分轨道多面体。《数学规划》114，1（2008），1–36。[15] Xavier Leroy、Damien Doligez、Alain Frisch、Jacques Garrigue、Didier Rémy 和 Jérôme Vouillon。2025。《OCaml系统》（5.4版）。法国国家信息与自动化研究所。可在 https://ocaml.org/manual/5.4/index.html 查阅；访问日期：2026年1月16日。[16] Leo Liberti 和 James Ostrowski。2014。数学规划问题的基于稳定子的对称性破除约束。《全球优化期刊》

60 2014 183 194 10 1007 10898 013 0106 6 ( ), –。doi: 。/s - - -

[17]1987《CPLEX用户手册》。。IBM ILOG CPLEX Optimization Studio。12版，1987–2018（1987），1。[18] François Margot。2002。分支割中的同构剪枝。《数学规划》94，1（2002），71–90。doi:10.1007/s10107-002-0358-2[19] François Margot。2010。整数线性规划中的对称性。载于《整数规划50年：1958–2008》，Michael Jünger、Thomas Liebling、Denis Naddef、George Nemhauser、William Pulleyblank、Gerhard Reinelt、Giovanni Rinaldi 和 Laurence Wolsey（编）。Springer，海德堡、多德雷赫特、伦敦、纽约，第17章，647–686。doi:10.1007/978-3-540-68279-0[20] Silvano Martello 和 Paolo Toth。1990。《背包问题：算法与计算机实现》。约翰·威立父子公司。[21] Aaron Meurer、Christopher P Smith、Mateusz Paprocki、Ondřej Čertík、Sergey B Kirpichev、Matthew Rocklin、AMiT Kumar、Sergiu Ivanov、Jason K Moore、Sartaj Singh 等。2017。SymPy：Python中的符号计算。《PeerJ计算机科学》3（2017），e103。[22] Christopher Muir、Luke Marshall 和 Alejandro Toriello。2024。半容量作业的时序装箱问题。《INFORMS优化期刊》6，1（2024），46–62。[23] Bruno R. Preiss。1998。《表达式树》。访问日期：2026年1月30日；https://web.archive.org/web/20170119094603/http://www.brpreiss.com/books/opus5/html/page264.html。[24] Jean-Francois Puget。1993。对称约束满足问题的可满足性。载于《智能系统方法论国际研讨会》。Springer，350–361。[25] Domenico Salvagnin。2018。基于Schreier-Sims表的对称性破缺不等式。载于《集成国际会议》

约束规划、人工智能与运筹学。Springer，521–529。[26] Toby Walsh。2012。对称性破除约束：最新结果。*arXiv* 预印本 *arXiv:1204.3348*（2012）。

详细的分类实验结果

本附录对实验结果进行了细粒度呈现。这些图所报告的数据与图 2相同；不过，结果按每个单独类别分别呈现。这些图证实了论文正文中所讨论的发现。

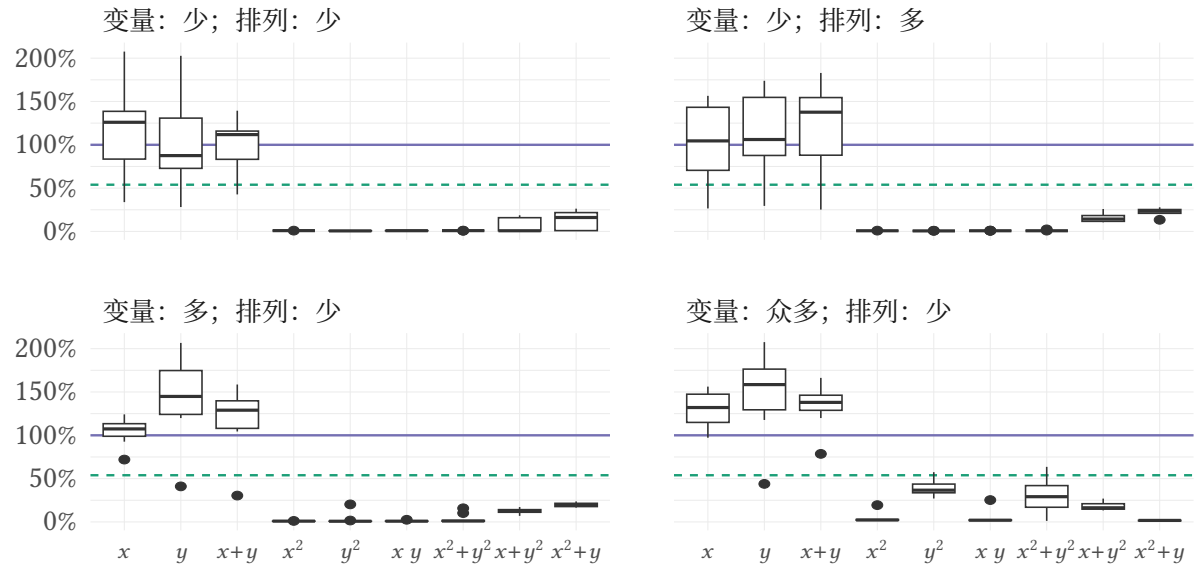

图3. 仅显示三类情形的结果。

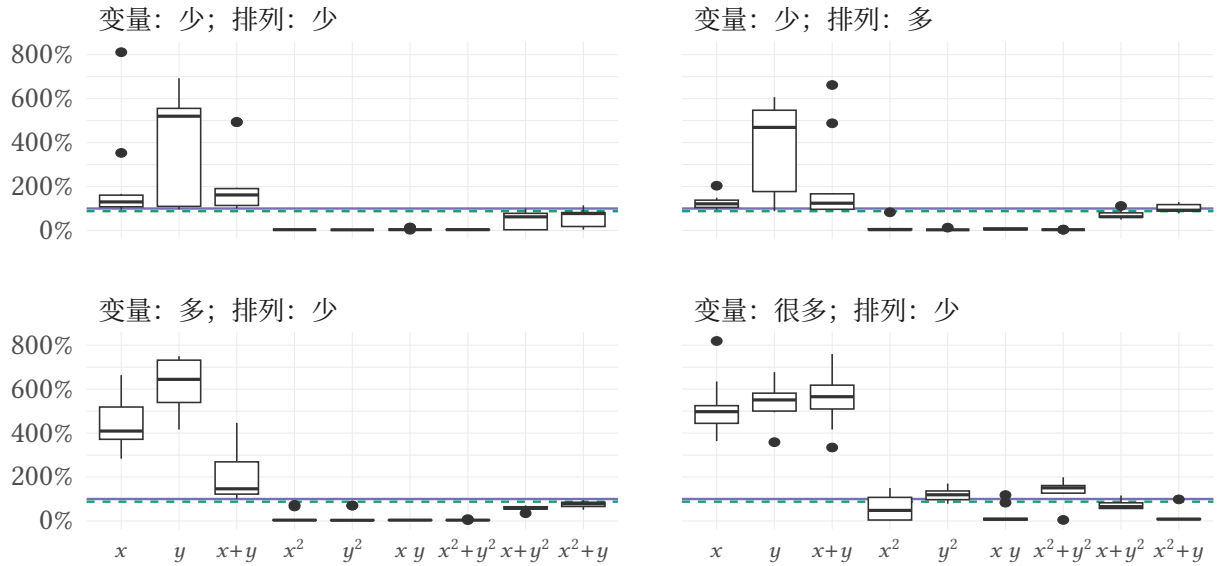

图4. 仅针对五类情况的结果。

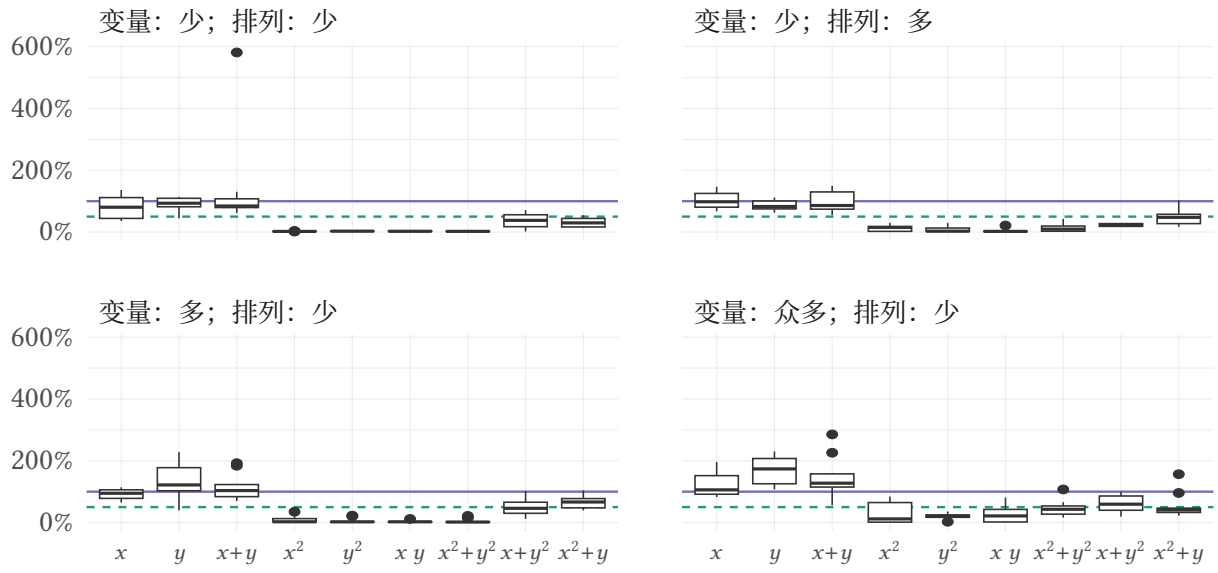

图5. 仅展示七类情况的结果。

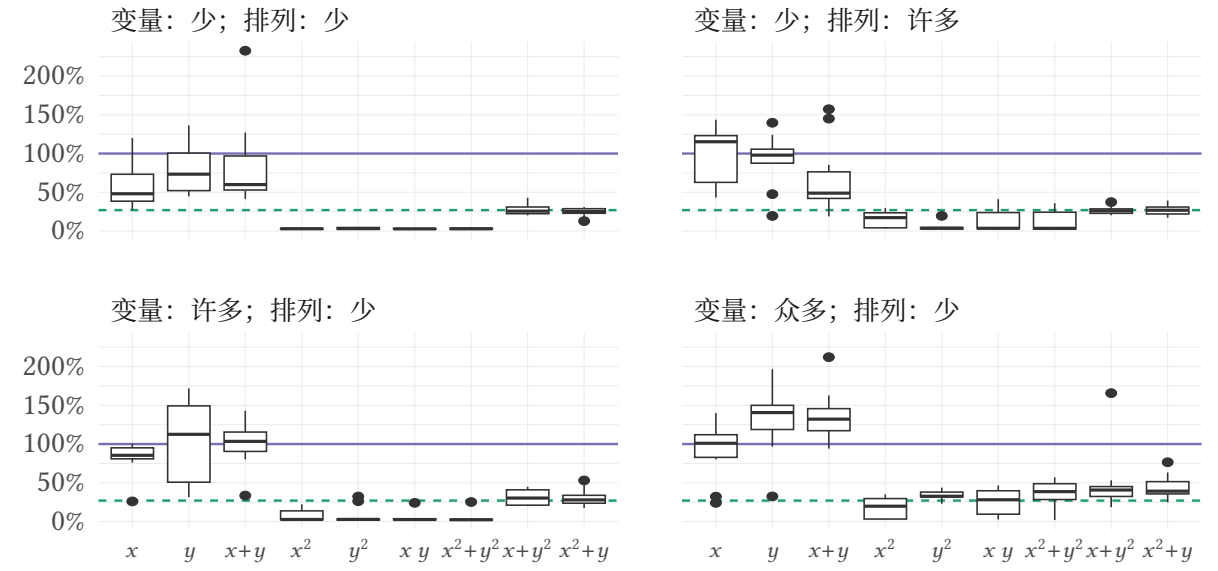

图6. 仅九类情况的结果。