Reductions that Consider Multiple Constraints and Variables Simultaneously

分享人：何华英

时间：2025.3.27

目录

- Presolve（常见的presolve形式）
- Reductions that Consider Multiple Constraints Simultaneously
  - Redundancy Detection（冗余约束检测）
  - Parallel and Nearly Parallel Rows（平行行检测）
  - Nonzero Cancellation（非零消除）
  - Bound and Coefficient Strengthening（界和系数强化）
  - Clique Merging（派系合并）
- Reductions that Consider Multiple Variables Simultaneously
  - Extension of Dual Fixing for Single Equations（单方程对偶固定的扩展）
  - Fix Redundant Penalty Variables（固定冗余惩罚变量）
  - Parallel Columns（平行列检测）
  - Dominated Columns（主导列检测）

1. MIP问题

一个MIP问题由目标、约束、决策变量组成

$$min \boldsymbol{c}_N^T \boldsymbol{x} + \boldsymbol{c}_Q^T \boldsymbol{y}$$

$$s.t. A_{iN}x + A_{iQ}y o_i b_i, \forall i \in M$$

约束函数

$$l \leq (x, y)^T \leq u$$

$$x \in Z_{\geq 0}^n, y \in R^q$$

决策变量域

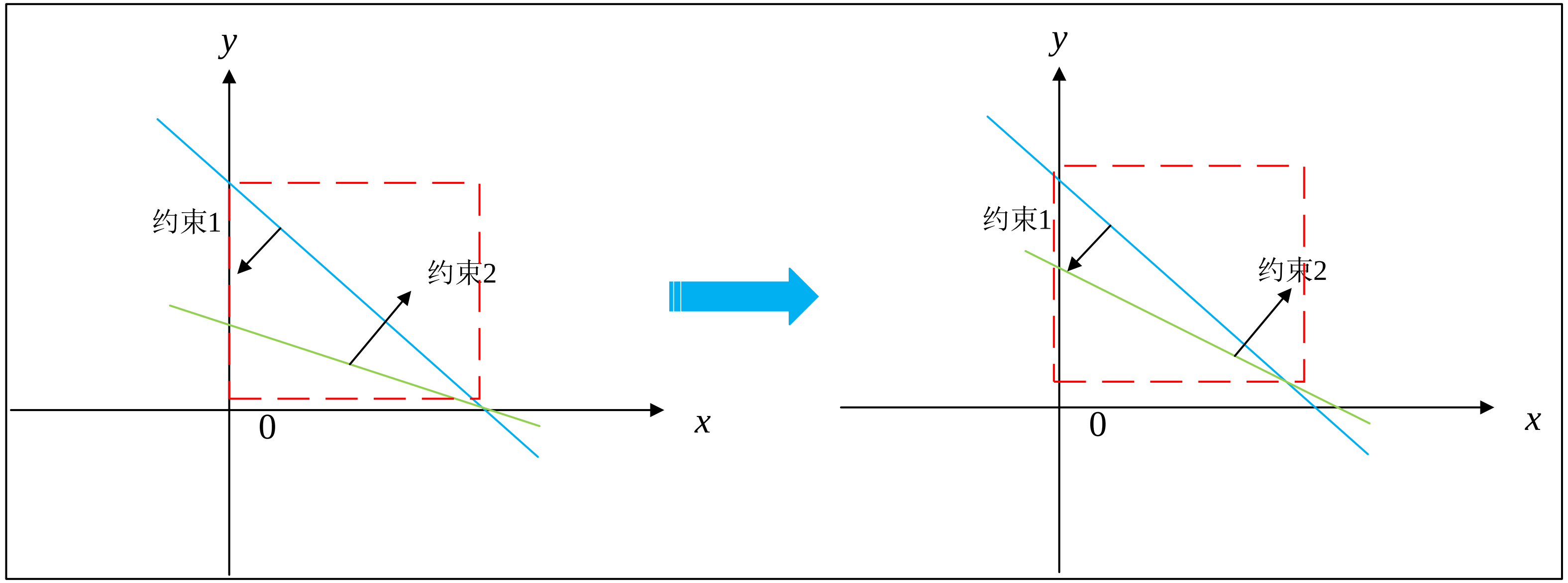
这两者共同决定模型规模的大小和可行域的空间大小

求解器进行求解时，更易于求解问题规模小（变量和约束少）、可行域范围小（变量范围小）的问题

如何缩减问题规模和可行域—presolve

1.1 什么是presolve

进行一系列操作，使得问题规模减小、变量的上下界更紧等



缩减了问题可行域

l <= 变量x1，x2，x3 <= u

l <= 变量x1，x3 <= u，x2=0

缩减了问题规模

1.2 几种基本的presolve

常见用于解决MIP问题的几种presolve形式

采用预处理技术，
旨在简化模型

- Coefficient tightening
- Domain propagation
- Colum singleton

⟹ **Fast**

- Dual fix
- Fix continuous
- Parallel columns
- Parallel rows
- Simple probing
- Stuffing

⟹ **Medium**

- Dominated column
- Implied integer
- Probing
- Sparsity
- substitution

⟹ **Exhaustive**

2. Reductions that Consider Multiple Constraints Simultaneously

求解器中同时考虑多个约束（行）的预处理方式，比单约束约简更复杂，需要限制工作量以避免过多无效计算

Table 14. Impact of Disabling All Multi-row Reductions

		Default	Disable multi-row reductions					Affected	
Bracket	Models	Tilim	Tilim	Faster	Slower	Time	Nodes	Models	Time
All	3,158	569	615	514	806	1.13	1.03	—	—
≥0 sec	2,616	32	78	514	806	1.17	1.09	2,013	1.23
≥1 sec	1,788	32	78	488	743	1.24	1.11	1,536	1.29
≥10 sec	1,266	32	78	346	597	1.34	1.12	1,116	1.40
≥100 sec	738	32	78	200	375	1.49	1.21	667	1.57
≥1,000 sec	286	32	78	74	158	2.02	1.41	272	2.10

禁用多行约简对求解时间较长的模型影响较大，求解时间和节点数的增加比例随着求解时间的增加而增加。多行约简在处理复杂模型时对性能的提升更为显著。

多行约简能够改进模型和运行时间

Redundancy Detection

Parallel and Nearly Parallel Rows

Nonzero Cancellation

Bound and Coefficient Strengthening

Clique Merging

2.1 Redundancy Detection

➢ **若删除某个约束后问题的可行解集不变，则该约束是冗余的**

$$\min_x c^T x$$

$$\text{s.t.} \sum_{j\in N} a_{ij}x_j \le b_i,\ \forall i \in M$$

$$l_j \le x_j \le u_j,\ x_j \in Z,\ \forall j \in N$$

*IP*问题

$$P_I = \{x \in Z^n | Ax \le b,\ l \le x \le u\}$$

可行域P_I

⇩ 删除约束$r \in M$

IP_r问题

IP_r问题可行域P_I^r

$P_I^r = P_I$，则称约束r冗余

判断冗余实质：研究对于线性**松弛**问题而言的冗余约束，即松弛问题冗余约束，考虑如下线性规划问题

$$\min_x c^T x \quad (1)$$

$$\text{s.t.} \sum_{j\in N} a_{ij}x_j \le b_i,\ \forall i \in M \quad (2)$$

$$l_j \le x_j \le u_j,\ x_j \in R,\ \forall j \in N \quad (3)$$

判断约束$r \in M$

$$\sum_{j\in N} a_{rj}x_{ij} \le b_r,\ r \in M \quad (4)$$

⇩

$$b_r^* = \max\left\{\sum_{j\in N} a_{rj}x_j \middle| \sum_{j\in N} a_{ij}x_{ij} \le b_i, \forall i \in M\setminus r,\ l \le x \le u,\ x \in R^n\right\} \quad (5)$$

当且仅当$b_r^* \le b_r$时，约束 (4) 对于线性规划问题 (1-3) 冗余

2.1 Redundancy Detection

➢ 若删除某个约束后问题的可行解集不变，则该约束是冗余的

□根据**变量上下界**来判断冗余约束 □根据**平行和近似平行行约束**判断冗余约束
□根据**线性相关方程**判断冗余约束 □根据**派系结构**判断冗余约束

Table 15. Impact of Enabling Dependent Row Detection

		Default	Enable dependent row detection					Affected	
Bracket	Models	Tilim	Tilim	Faster	Slower	Time	Nodes	Models	Time
All	3,175	569	575	195	203	1.02	1.02	—	—
≥0 sec	2,615	13	19	195	203	1.02	1.02	737	1.07
≥1 sec	1,754	13	19	188	201	1.03	1.04	602	1.09
≥10 sec	1,196	13	19	151	147	1.04	1.04	430	1.11
≥100 sec	644	13	19	95	86	1.05	1.05	232	1.17
≥1,000 sec	193	13	19	34	47	1.25	1.29	93	1.65

Dependent Row Detection

Table 16. Impact of Enabling Clique Subsumption Detection

		Default	Enable clique subsumption detection					Affected	
Bracket	Models	Tilim	Tilim	Faster	Slower	Time	Nodes	Models	Time
All	3,178	571	565	144	164	1.00	1.01	—	—
≥0 sec	2,614	12	6	144	164	1.00	1.02	553	1.01
≥1 sec	1,751	12	6	142	162	1.00	1.02	496	1.01
≥10 sec	1,191	12	6	113	131	1.01	1.01	364	1.01
≥100 sec	629	12	6	78	79	1.00	1.01	217	1.00
≥1,000 sec	173	12	6	27	35	1.01	1.03	80	1.03

Inequality Subsumption Detection

Table 17. Impact of Enabling Inequality Subsumption Detection

		Default	Enable inequality subsumption detection					Affected	
Bracket	Models	Tilim	Tilim	Faster	Slower	Time	Nodes	Models	Time
All	3,179	571	572	184	194	1.00	1.01	—	—
≥0 sec	2,616	13	14	184	194	1.00	1.01	637	1.00
≥1 sec	1,751	13	14	181	192	1.00	1.02	564	1.00
≥10 sec	1,197	13	14	150	168	1.01	1.02	439	1.02
≥100 sec	633	13	14	102	107	1.01	1.03	269	1.01
≥1,000 sec	187	13	14	35	47	1.07	1.14	102	1.14

Clique Subsumption Detection

2.2 Parallel and Nearly Parallel Rows

➢ **平行行：指两个约束的系数向量成比例，此时可以删除其中一个约束**

对于两个约束 q 和 r，若存在标量 s 使得 $\boldsymbol{sa_r = a_q}$，则它们是平行的

□ q 和 r都为等式约束

$$\sum_{j \in N} a_{rj}x_j = b_r$$
$$\sum_{j \in N} a_{qj}x_j = b_q$$

- 若$sb_r = b_q$，则这两个约束等价，可以删除其中任意一个；
- 若$sb_r \neq b_q$ 则说明该问题不可行

$$3x + 4y = 11 \ (1)$$
$$0.6x + 0.8y = 2.2(2)$$

删除

□ q 为等式，r为不等式约束

$$\sum_{j \in N} a_{rj}x_j \leq b_r$$
$$\sum_{j \in N} a_{qj}x_j = b_q$$

- 若$b_q \leq sb_r$，满足约束q则一定满足约束r，约束r冗余可以删除；
- 若$b_q > sb_r$，约束q与约束r矛盾，该问题不可行

$$3x + 4y \leq 11 \ (3)$$
$$0.6x + 0.8y = 2(4)$$
$$0.6x + 0.8y = 3(5)$$

删除

不可行

□ q 和r均为不等式约束

$$\sum_{j \in N} a_{rj}x_j \leq b_r$$
$$\sum_{j \in N} a_{qj}x_j \leq b_q$$

- 若$b_q \leq sb_r$，满足约束q则一定满足约束r，约束r冗余可以删除；
- 若$b_q > sb_r$，满足约束r则一定满足约束q，约束q冗余可以删除；

$$3x + 4y \leq 11 \ (6)$$
$$0.6x + 0.8y \leq 2(7)$$
$$0.6x + 0.8y \leq 3(8)$$

删除

删除

2.2 Parallel and Nearly Parallel Rows

- 近似平行行：指两个约束成比例，或其系数向量成比例，此时可以删除其中一个约束

$$y + 3x_1 + 4x_2 + 6x_3 \le 11\ (1)$$
$$z + 0.3x_1 + 0.4x_2 + 0.6x_3 \le 6(2)$$

约束（1）和约束（2）为近似平行约束，这两个约束中只有一个变量不一样就是y, z，且约束（1）中其他的变量系数是约束（2）中其他变量系数的10倍

$$a_{q1}x_1 + a_{qc}x_c = b_q\ (3)$$
$$a_{r2}x_2 + a_{rc}x_c = b_r(4)$$

其中$a_{r2} \neq 0$，若$sa_{qc} = a_{rc}$则表明约束（3）和约束（4）为近似平行约束。其中x_c是进行平行约束中的公共变量，x_1和x_2分别为约束（3）和约束（4）的独有变量

将约束（3）乘以s，再减去约束（4）可得

$$sa_{q1}x_1 - a_{r2}x_2 = b_q s - b_r$$

即x2可以用x1表示

$$x_2 = tx_1 + d\ (5)$$
$$t = \frac{sa_{q1}}{a_{r2}}, d = (b_r - b_q s)/\ a_{r2}$$

约束（5）带入约束（4）

$$sa_{q1}x_1 + a_{rc}x_c = b_q s\ (6)$$

将近似平行约束（1）和约束（2）等价转化为平行约束（3）和约束（6），随后采用上一节所述处理平行约束的方法来处理

2.3 Nonzero Cancellation

➢ **非0元素消除：通过将一个方程加到其他约束中，减少系数矩阵中的非零元素的数量**

将方程添加到其他约束中，可以得到一个等价模型。

$$
\begin{aligned}
A_{qS}x_S + A_{qT}x_T + A_{qU}x_U \quad &= b_q \\
A_{rS}x_S + A_{rT}x_T \quad + A_{rV}x_V &\le b_r
\end{aligned}
$$

⟹

假设存在一个标量s，对于所有$j \in T$，有$sA_{qs} = A_{rs}, sA_{qj} \neq A_{rj}$那么，从行$r$中减去$s$倍的行$q$：

$$
\begin{aligned}
A_{qS}x_S + A_{qT}x_T \quad + A_{qU}x_U \quad &= b_q \\
(A_{rT} - sA_{qT})x_T - sA_{qU}x_U + A_{rV}x_V &\le b_r - sb_q.
\end{aligned}
$$

矩阵中的非零系数数量减少了$|S|-|U|$，因此如果$|S|>|U|$，应该进行这种变换

$$
\begin{aligned}
2x_1 + 3x_2 + 4x_3 &= 12 \ (1) \\
4x_1 + 6x_2 + 5x_4 &\le 24 \ (2)
\end{aligned}
$$

⟹ 约束（2）- 约束（1）*2

$$
\begin{aligned}
2x_1 + 3x_2 + 4x_3 &= 12 \ (1) \\
5x_4 - 8x_3 &\le 0 \ (3)
\end{aligned}
$$

x_S x_V

通过这个操作，非0系数向量减少了2-1=1个

2.4 Bound and Coefficient Strengthening

➢ **通过计算约束中变量的上下界，强化变量的界和系数**

对于某约束，通过推导出某些变量或表达式的更紧的上界/下界，从而收紧变量的取值范围（边界强化）和减小线性约束中某些变量的系数（系数强化）

$$\boxed{A_{iS}x_S + a_{ik}x_k \le b_i}$$

通过分析$A_{iS}x_S$的取值范围，来推导出对x_k的限制，甚至进一步地优化 x_k的系数a_{ik}

⇩

$$\boxed{l_{iS} \le A_{iS}x_S \le u_{iS}}$$

理想方法

$$l_{iS} = \min\{A_{iS}x_S: x \in P_{MIP}\}, u_{iS} = \max\{A_{iS}x_S: x \in P_{MIP}\}$$

相当于每1个不等式都要进行2个MIP问题（满足原来约束）求解，非常耗时

改进方法

$$l_{iS} = \min\{A_{iS}x_S: x \in P_{LP}\}, u_{iS} = \max\{A_{iS}x_S: x \in P_{LP}\}$$

转化为求解2个LP问题，但是耗时还是很长

2.4 Bound and Coefficient Strengthening

➢ **通过计算约束中变量的上下界，强化变量的界和系数**

方法1

$$A_{iS_1}x_{S_1} + \ldots + A_{iS_d}x_{S_d} + a_{ik}x_k \le b_i$$

分块计算（block partition）：

- 把变量集合x_S分成多个子块$s_1, s_2, .., s_d$，分别计算每块的上下界。
- 然后把各个子块的上下界加总起来得到整体上下界。

- 每一项求上下界$l_{is_p} = \inf\{A_{is_p}x_{s_p}\}, u_{is_p} = \sup\{A_{is_p}x_{s_p}\}$；
- 各个子块的上下界相加，$l_{iS} = l_{is_1} + \cdots + l_{is_d}, u_{iS} = u_{is_1} + \cdots + u_{is_d}$；
- 改写约束：$a_{ik}x_k \le b_i - l_{iS}$，$x_k \le (b_i - l_{iS})/a_{ik}$；
- 调整系数：若x_k是整数，可利用上下界调整a_{ik}。

方法2

$$\min\{x_k : x \in P_{\text{MIP}}\} \le x_k \le \max\{x_k : x \in P_{\text{MIP}}\}.$$

基于优化的边界收紧（OBBT）：通过求解（简化的）优化问题，来获得变量可能的最小值和最大值，来收紧变量界限。

$$\boxed{\begin{array}{c} 2x_1 + 3x_2 + x_3 \le 12 \ (1) \\ 0 \le x_1, x_2 \le 2, x_3 \ge 0 \end{array}}$$

由于x_1和x_2自身范围，很容易得到约束 (1)除去x_3的边界为（$l_{is} = 0, u_{is} = 10$），利用边界对x_3进行上下界强化，得到$0 \le x_3 \le 2$

通过这个操作，进一步强化了x_3的上下界

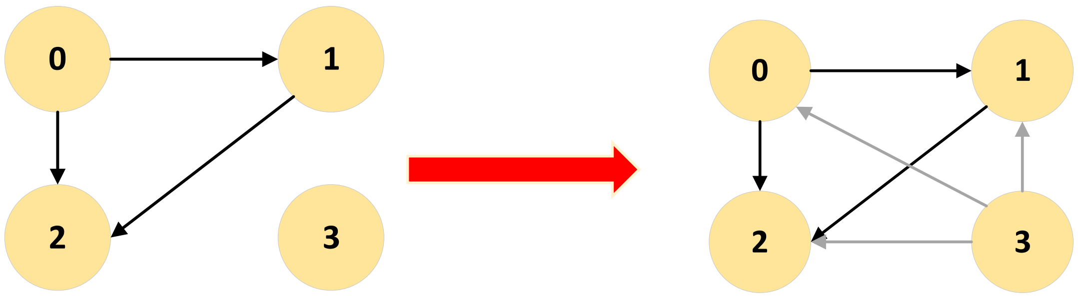
2.5 Clique Merging

➢ 识别模型中的派系结构，将多个约束合并为一个更强的约束

将多个类似的约束合并成一个更强（通用）的约束，从而减少约束的数量并增强模型的表达

$$\sum_{j\in S} x_j + \sum_{j\in T}(1 - x_j) \le 1\ (= 1)$$

其中$S, T \subset I, S \cap T = \emptyset$，且对所有$j \in S \cup T$来说，$l_j = 0, u_j = 1$，会产生$|S \cup T| \cdot (|S \cup T| - 1)/2$条边



Clique Extension：图中已经知道一个 Clique 是$\{0,1,2\}$，即$x_0 + x_1 + x_2 \le 1$。若想扩展这个 Clique，从不在 Clique 的点从挨个去试，如果有一个点和当前 Clique 内的所有的点都是连接在一起的，把这个点也加入到当前的 Clique 中。比如尝试将点3加入当前clique，也成立，即获得$x_0 + x_1 + x_2 + x_3 \le 1$

$$\boxed{\begin{aligned} x_1 + x_2 \le 1 \\ x_1 + x_3 \le 1 \\ x_2 + x_3 \le 1 \end{aligned}}$$

将3个约束合并为1个更强且通用的约束

$$\boxed{x_1 + x_2 + x_3 \le 1}$$

通过这个操作，3个约束转化为1个约束

3. Reductions that Consider Multiple Variables Simultaneously

与上一章节相对应，求解器中同时考虑多个变量的预处理方式，比固定单一变量更复杂

Table 22. Impact of Disabling All Multi-column Reductions

		Default	Disable multi-column reductions					Affected	
Bracket	Models	Tilim	Tilim	Faster	Slower	Time	Nodes	Models	Time
All	3,173	570	572	337	459	1.04	1.02	—	—
≥0 sec	2,616	18	20	337	459	1.05	1.04	1,454	1.09
≥1 sec	1,770	18	20	326	441	1.07	1.05	1,135	1.12
≥10 sec	1,219	18	20	252	340	1.09	1.06	819	1.14
≥100 sec	666	18	20	144	211	1.12	1.08	462	1.17
≥1,000 sec	210	18	20	54	79	1.21	1.14	166	1.25

多列约简能够改进模型和运行时间

禁用多列约简会导致求解时间的几何平均值在所有模型分类中均有所增加，同时节点数的几何平均值也有所上升，这表明多列约简在求解混合整数规划问题时对性能有积极的影响，尤其在处理较难的模型时更为明显，但相比多行约简效果差一些

Fix Redundant Penalty Variables

Parallel Columns

Dominated Columns

Extension of Dual Fixing for Single Equations

3.1 Extension of Dual Fixing for Single Equations

➢ **若删除某列（变量）后问题的可行解集不变，则该列（变量）是冗余的**

如果问题有特定的结构，则可以用**对偶固定**的**扩展**将**变量固定到其边界值**来简化模型。它适用于变量仅在一个方程中的情况，当方程中有连续变量时，这些变量可补偿被固定的变量，从而保持模型的可行性和最优性。

$$
\begin{aligned}
\min\ & x_1 + x_2 + x_3 \\
& 2x_1 + 4x_2 - 3x_3 \le 8 \\
& x_1 - x_2 - 2x_3 \le -4 \\
& 0 \le x_1 \le 2,\ x_2, x_3 \in \{0,1\}
\end{aligned}
$$

对于变量x_1而言，它的目标函数系数是正的，同时在所有小于等于的约束里x_1的系数都是正的。由此易得知：越小的x_1其约束左端项的值也越小，更有利于约束被满足；目标函数系数是正的，所以目标函数希望x_1越小越好。基于以上两点，，可以把x_1固定到下界，也就得到 $x_1 = 0$，这就是对偶固定（Dual Fixing）算法。

⇩

$$
\begin{aligned}
\min_{x}\ & c^T x \\
\text{s.t.}\ & \sum_{j\in N} a_{ij}x_j \le b_i,\ \forall i \in M \\
& l_j \le x_j \le u_j,\ x_j \in Z,\ \forall j \in N
\end{aligned}
$$

□ $c_j \ge 0, a_{ij} \ge 0, \forall i \in M$

$c_j \ge 0, a_{ij} \ge 0$，x_j越小越好，x_j变小会使得所有的约束变得更加可行，由此可以**把x_j固定为l_j**。若l_j是负无穷，则说明该问题是不可行或者无界。

□ $c_j \le 0, a_{ij} \le 0, \forall i \in M$

与第一种情况类似，因此将 **x_j 固定为 u_j**。若u_j是正无穷，则说明该问题是不可行或无界。

3.2 Fix Redundant Penalty Variables

➢ **提前识别并固定一些"不会被使用"的惩罚变量为 0（适用条件严苛）**

识别模型中的**惩罚变量**，并在可能的情况下将其**固定为下界**

$$\begin{aligned} \min\ & c_P^T x_P + c_V^T x_V \\ & A_{rP} x_P + A_{rV} x_V \le b_r \\ & A_{TV} x_V \le b_T \end{aligned}$$

其中，对于所有$j \in I, x_j \in Z$，$l \le x \le u$，
$c_p > 0, A_{rP} < 0$（惩罚变量成本系数+，约束系数-）

⇩

$$\frac{c_1}{|a_{r1}|} \le \cdots \le \frac{c_p}{|a_{rp}|}.$$ 惩罚变量的成本逐渐增加

$$\sup\{A_{rV} x_V\} + \sum_{t=1}^{k} a_{rt} u_t + \sum_{t=k+1}^{p} a_{rt} \ell_t \le b_r, \quad \textbf{(0)}$$

则可以将$x_{k+1}, .., x_p$分别固定为$l_{k+1}, \dots, l_p$

$$\begin{gathered} \min 2x_1 + 3x_2 + 5y \ (1) \\ 6 - x_1 - x_2 \le 5 \ (2) \\ y \ge 10 \ (3) \\ x_1, x_2, y \ge 0 \\ \text{其中} x_1, x_2 \text{为惩罚变量} \end{gathered}$$

⇩

- 按照单位惩罚成本排序：对于x_1，$\frac{2}{-(-1)} = 2$；对于x_2，$\frac{3}{-(-1)} = 3$。所以排序为：$x_1 \le x_2$。
- 固定较贵的惩罚变量：用约束（0）检查是否可以固定x_2。
- 评估约束条件：仅使用x_1时，约束变为$6 - x_1 \le 5$，即$x_1 \ge 1$。就可以固定/移除惩罚变量x_2 。
- 简化模型：

$$\begin{gathered} \min 2x_1 + 5y \ (1) \\ x_1 \ge 1 \ (2) \\ y \ge 10 \ (3) \end{gathered}$$

通过这个操作，固定了x_2，缩紧了x1的下界

3.3 Parallel Columns

➢ **平行列是指两个变量的系数向量成比例**

对于两个约束 j 和 k，若存在标量 s 使得 $A_j = sA_k$ ，则它们是平行的，可以高效地检测平行**列**

$$\min_x c^T x$$

$$\text{s.t.} \sum_{j\in N} a_{ij}x_j \le b_i,\ \forall i \in M$$

$$l_j \le x_j \le u_j,\ x_j \in Z,\ \forall j \in N$$

采用 a_i 来表示约束矩阵第 i 列元素构成的向量，那么对于两个变量x_i和x_j若满足如下条件则称两个变量为平行变量

$$\boxed{\begin{gathered} a\_i = \lambda a\_j \\ c_i = \lambda c_j \end{gathered}}$$

$$\Downarrow$$

$$\boxed{y := x_i + \lambda x_j}$$

$$l_y := l_i + \lambda l_j$$

$$u_y := u_i + \lambda u_j$$

$$c_y := c_i$$

$$\min 6x_1 + 3x_2 + x_3 \quad (1)$$

$$\text{s.t. } 2x_1 + x_2 + x_3 \le 6 \quad (2)$$

$$4x_1 + 2x_2 + 5x_3 + x_4 \le 10 \quad (3)$$

$$x_1, x_2, x_3, x_4 \in \{0,1\} \quad (4)$$

$$\begin{pmatrix} 2 \\ 4 \end{pmatrix} = 2\begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$\Downarrow$$

$$y := 2x_1 + x_2$$

$$\min 3y + x_3$$

$$\text{s.t. } y + x_3 \le 6$$

$$2y + 5x_3 + x_4 \le 10$$

$$0 \le y \le 3, y \in Z, x_3, x_4 \in \{0,1\}$$

通过这个操作，把平行变量合并成了一个变量，减少了**变量的数量**

3.4 Dominated Columns

➢ 分析变量之间的支配关系，推导出变量的固定或替代方案

定义**变量之间**的**支配**关系，若一个变量在所有约束中的**系数**和**目标函数值**都**小于**等于另一个变量，则前者支配后者，如果两个变量的上下界及其整数性满足特定条件，就可以推断出变量的固定值。

$$\min_x c^T x$$
$$\text{s.t.} \sum_{j\in N} a_{ij}x_j \le b_i,\ \forall i \in M$$
$$l_j \le x_j \le u_j,\ x_j \in Z,\ \forall j \in N$$

对于变量x_i和x_j，满足下列条件则称x_i支配x_j：

$$a_{ik} \le a_{jk} \quad \forall i \in M$$
$$c_i \le c_j$$

➢ **处理思路：**

- 构造新解：假设存在一个可行解$\hat{x} = (\hat{x}_i, \hat{x}_j)$，则$x^* = (\hat{x}_i + \alpha, \hat{x}_j - \alpha)(\alpha > 0)$也肯定是一个可行解。
- 检查滑动是否有利：原解目标函数为$c^T\hat{x} = c_i\hat{x}_i + c_j\hat{x}_j$；变为$c_i(\hat{x}_i + \alpha) + c_j(\hat{x}_j - \alpha) = c^T\hat{x} + \alpha(c_i - c_j) \le c^T\hat{x}$，变小。
- 滑动受边界的限制：$l_i \le x_i \le u_i, l_j \le x_j \le u_j$，所以$\alpha \le \min(u_i - \hat{x}_i, \hat{x}_j - l_j)$。
- 固定值的取值：所以直到$\boldsymbol{\alpha = \min(u_i - \hat{x}_i, \hat{x}_j - l_j)}$，滑倒边界，$x_i = u_i$或$x_j = l_j$时达到了目标函数最优所能到达的支配方向终点。

3.4 Dominated Columns

➢ 分析变量之间的支配关系，推导出变量的固定或替代方案

定义**变量之间**的**支配**关系，若一个变量在所有约束中的**系数**和**目标函数值**都**小于**等于另一个变量，则前者支配后者，如果两个变量的上下界及其整数性满足特定条件，就可以推断出变量的固定值。

x^*必定在可行域内，所以$\hat{x}_i+\alpha$和$\hat{x}_j-\alpha$也要满足可行域约束，即

$$l_i \le \hat{x}_i+\alpha \le u_i,\quad l_j \le \hat{x}_j-\alpha \le u_j$$

解得

$$\alpha \in [\max(l_i-\hat{x}_i, \hat{x}_j-u_j), \min(u_i-\hat{x}_i, \hat{x}_j-l_j)]$$

由于α的增大有利于目标函数值，因此$\boldsymbol{\alpha=min(u_i-\hat{x}_i, \hat{x}_j-l_j)}$

特殊情况处理

- x_i的上界u_i是正无穷大， x_i支配x_j， 则x_j被固定为l_j；
- x_i的上界u_i是正无穷大， x_i支配$-x_j$， 则x_j被固定为u_j；
- x_j的下界l_j是负无穷大， x_i支配x_j， 则x_i被固定为u_i；
- x_j的下界l_j是负无穷大， $-x_i$支配x_j， 则x_i被固定为l_i。

$$\min x_1+2x_2 \ (1)$$
$$x_1+2x_2 \le 10 \ (2)$$
$$x_1 \ge 2 \ (3)$$
$$-5 \le x_2 \le -1 \ (4)$$

此时 $\alpha=min(\infty-\widehat{x_1}, \widehat{x_2}+5)$， 而由于 x_2 的范围是负数，α越大越好，所以只有在$x_2=-1$时才是更好的固定。为了简便计算，记x_i支配$-x_j$， 上下界作同等替换。

3.4 Dominated Columns

➢ 分析变量之间的支配关系，推导出变量的固定或替代方案

定义**变量之间**的**支配**关系，若一个变量在所有约束中的**系数**和**目标函数值**都**小于**等于另一个变量，则前者支配后者，如果两个变量的上下界及其整数性满足特定条件，就可以推断出变量的固定值。

$$\max 3x_1 + 5x_2 \ (1)$$
$$2x_1 + 4x_2 \le 100 \ (2)$$
$$x_1 \le 30 \ (3)$$
$$x_2 \le 25 \ (4)$$
$$x_1, x_2 \ge 0$$

- 识别支配关系： x_1支配x_2。
检查定理条件： x_1和x_2均为有界上下界，常规处理$\alpha = min(u_1 - \widehat{x_1}, \widehat{x_2} - l_1)$，可以取$x_1$的上界或$x_2$的下界 。
- 固定变量：x_1支配x_2，随机选取1个变量进行固定，固定x_2为其下界，即$x_2 = l_2 = 0$，即可移除x_2，缩紧x_1 。
- 调整后的模型：

$$\max 3x_1 \ (1)$$
$$2x_1 \le 100 \ (2)$$
$$x_1 \le 30 \ (3)$$
$$x_1 \ge 0$$

通过这个操作，固定了1个变量，且缩进了边界上下界

Thank You !