DM872
Math Optimization at Work

Preprocessing

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline

Outline

Preprocessing rules

Consider $S = \{\boldsymbol{x} : a_0 x_0 + \sum_{j=1}^n a_j x_j \leq b, l_j \leq x_j \leq u_j, j = 0..n\}$

- Bounds on variables.
  If $a_0 > 0$ then:

$$x_0 \leq \left(b - \sum_{j:a_j>0} a_j l_j - \sum_{j:a_j<0} a_j u_j \right) / a_0$$

  and if $a_0 < 0$ then

$$x_0 \geq \left(b - \sum_{j:a_j>0} a_j l_j - \sum_{j:a_j<0} a_j u_j \right) / a_0$$

- Redundancy. The constraint $\sum_{j=0}^n a_j x_j \leq b$ is redundant if

$$\sum_{j:a_j>0} a_j u_j + \sum_{j:a_j<0} a_j l_j \leq b$$

- Infeasibility: $S = \emptyset$ if (swapping lower and upper bounds from previous case)

$$\sum_{j:a_j>0} a_j l_j + \sum_{j:a_j<0} a_j u_j > b$$

- Variable fixing. For a max problem in the form

$$\max\{\boldsymbol{c}^T\boldsymbol{x} : A\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}\}$$

 if $\forall i = 1..m : a_{ij} \geq 0, c_j < 0$ then fix $x_j = l_j$
 if $\forall i = 1..m : a_{ij} < 0, c_j > 0$ then fix $x_j = u_j$

- Integer variables:

$$\lceil l_j \rceil \leq x_j \leq \lfloor u_j \rfloor$$

- Binary variables. Probing: add a constraint, eg, $x_2 = 0$ and check what happens

Example

$$
\begin{aligned}
\max\ & 2x_1 + x_2 - x_3 \\
\mathtt{R1}: & 5x_1 - 2x_2 + 8x_3 \le 15 \\
\mathtt{R2}: & 8x_1 + 3x_2 - x_3 \ge 9 \\
\mathtt{R3}: & x_1 + x_2 + x_3 \le 6 \\
& 0 \le x_1 \le 3 \\
& 0 \le x_2 \le 1 \\
& x_3 \ge 1
\end{aligned}
$$

$$\mathtt{R1}: 5x_1 \le 15 + 2x_2 - 8x_3 \le 15 + 2 \cdot \overbrace{1}^{u_2} - 8 \cdot \overbrace{1}^{l_3} = 9 \qquad \rightsquigarrow x_1 \le 9/5$$

$$8x_3 \le 15 + 2x_2 - 5x_1 \le 15 + 2 \cdot 1 - 5 \cdot 0 = 17 \qquad \rightsquigarrow x_3 \le 17/8$$

$$2x_2 \ge 5x_1 + 8x_3 - 15 \ge 5 \cdot 0 + 8 \cdot 1 = -7 \qquad \rightsquigarrow x_2 \ge -7/2, x_2 \ge 0$$

$$\mathtt{R2}: 8x_1 \ge 9 - 3x_2 + x_3 \ge 9 - 3 + 1 = 7 \qquad \rightsquigarrow x_1 \ge 7/8$$

$$\mathtt{R1}: 8x_3 \ge 15 + 2x_2 - 5x_1 \le 15 + 2 - 5 \cdot 7/8 = 101/8 \qquad \rightsquigarrow x_3 \le 101/64$$

$\mathtt{R3}: x_1 + x_2 + x_3 \le 9/5 + 1 + 101/64 < 6$ Hence $\mathtt{R3}$ is redundant

Example

$$\begin{aligned} &\max\ 2x_1 + x_2 - x_3 \\ &\texttt{R1}: 5x_1 - 2x_2 + 8x_3 \le 15 \\ &\texttt{R2}: 8x_1 + 3x_2 - x_3 \ge 9 \\ &\quad 7/8 \le x_1 \le 9/5 \\ &\quad 0 \le x_2 \le 1 \\ &\quad 1 \le x_3 \le 101/64 \end{aligned}$$

Increasing x_2 makes constraints satisfied $\rightsquigarrow$ $x_2 = 1$
Decreasing x_3 makes constraints satisfied $\rightsquigarrow$ $x_3 = 1$

We are left with:

$$\max\{2x_1 : 7/8 \le x_1 \le 9/5\}$$

Preprocessing for Set Covering/Partitioning

1. if $e_i^T A = 0$ then the ith row can never be satisfied

$$\begin{bmatrix} 0 & 0 & \dots & 1 & \dots & 0 \end{bmatrix} \begin{bmatrix} & & & \\ \hdashline 0 \dots 0 & & \dots & 0 \\ \hdashline & & & \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

2. if $e_i^T A = e_k$ then $x_k = 1$ in every feasible solution

$$\begin{bmatrix} 0 & 0 & \dots & 1 & \dots & 0 \end{bmatrix} \begin{bmatrix} & & & & \\ \hdashline 0 & \dots & 1 & \dots & 0 \\ \hdashline & & & & \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

In SPP can remove all rows t with $a_{tk} = 1$ and set $x_j = 0$ (ie, remove cols) for all cols that cover t

3. if $e_t^T A \geq e_p^T A$ then we can remove row t, row p dominates row t (by covering p we cover t)

$$
\begin{array}{c}
t \\ \\ p
\end{array}
\begin{bmatrix}
 & 1 & 1 & 1 & \\
 & & & & \\
 & 1 & & 1 &
\end{bmatrix}
$$

In SPP we can remove all cols j: $a_{tj} = 1, a_{pj} = 0$

4. if $\sum_{j \in S} Ae_j = Ae_k$ and $\sum_{j \in S} c_j \leq c_k$ then we can cover the rows by Ae_k more cheaply with S and set $x_k = 0$
(Note, we cannot remove S if $\sum_{j \in S} c_j \geq c_k$)

$$
\begin{bmatrix}
 & & 1 & & & 1 \\
 & 1 & & & & 1 \\
 & & & 1 & & 1 \\
 & 0 & 0 & 0 & & 0 \\
 & 1 & & & & 1 \\
 & 0 & 0 & 0 & & 0
\end{bmatrix}
$$

Hot Topic

- MIP challenge 2024 on Presolving Reductions `https://github.com/dominiqs81/MIPcc24`
- Papilo, parallel presolving `https://github.com/dominiqs81/MIPcc24`

Summary

1. Preprocessing