Parametric programming: **Week 8 Lecture Notes**

$$\begin{aligned} \text{minimize} \quad & (\mathbf{c} + \theta \mathbf{d})'\mathbf{x} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

Solve for every value of θ

Example:

$$\begin{aligned} \text{minimize} \quad & (-3 + 2\theta)x_1 + (3 - \theta)x_2 + x_3 \\ \text{subject to} \quad & x_1 + 2x_2 - 3x_3 + x_4 = 5 \\ & 2x_1 + x_2 - 4x_3 + x_5 = 7 \\ & x \geq 0 \end{aligned}$$

Optimal cost:

$$g(\theta) = \min_{i=1,\ldots,N} (\mathbf{c} + \theta \mathbf{d})'\mathbf{x}^i,$$

$\mathbf{x}^1, \ldots, \mathbf{x}^N$ are the extreme points of the feasible set

(Parametric) simplex tableau

0	$-3+2\theta$	$3-\theta$	1	0	0
5	1	2	-3	1	0
7	2	1	-4	0	1

- If $-3 + 2\theta \geq 0$ and $3 - \theta \geq 0$, all reduced costs are nonnegative and we have an optimal basic feasible solution.

$$g(\theta) = 0, \qquad \frac{3}{2} \leq \theta \leq 3.$$

- For $\theta > 3$, have x_2 enter the basis
- New tableau:

$-7.5+2.5\theta$	$-4.5+2.5\theta$	0	$5.5-1.5\theta$	$-1.5+0.5\theta$	0
2.5	0.5	1	-1.5	0.5	0
4.5	1.5	0	-2.5	-0.5	1

- All reduced costs nonnegative if $3 \leq \theta \leq 5.5/1.5$
- Optimal cost

$$g(\theta) = 7.5 - 2.5\theta, \qquad 3 \leq \theta \leq \frac{5.5}{1.5}$$

- For $\theta > 5.5/1.5$, reduced cost of x_3 is negative.
- No positive pivot element
- For $\theta > 5.5/1.5$, $g(\theta) = -\infty$
- Proceed similarly for $\theta < 3/2$

Parametric programming more generally

- Reduced costs depend linearly on θ

- Bfs and basis matrix $\mathbf{B}$, optimal for $\theta_1 \leq \theta \leq \theta_2$

- Reduced cost of x_j negative for $\theta > \theta_2$.
  - Reduced cost is zero for $\theta = \theta_2$

- If $\mathbf{B}^{-1}\mathbf{A}_j \leq \mathbf{0}$, $g(\theta) = -\infty$ for $\theta > \theta_2$.

- Otherwise, bring x_j into basis

- Still have optimal solution at $\theta = \theta_2$.
- Range of θ under which new basis is optimal $[\theta_2, \theta_3]$
- If $\theta_i < \theta_{i+1}$, no basis repeated twice
- Change of basis: breakpoints of $g(\theta)$
- If $\theta_i = \theta_{i+1}$, method may cycle

Dual parametric programming

- Keep $\mathbf{c}$ fixed
- Right–hand side $\mathbf{b} + \theta\mathbf{d}$
- If increasing θ makes a basic variable negative, do a dual simplex iteration

Delayed column generation

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}'\mathbf{x} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

- $\mathbf{A}$ has a huge number of columns
  Can't form $\mathbf{A}$ explicitly
- All that simplex needs is to discover i with $\overline{c}_i < 0$ when one exists
- Assume we can solve the problem:

  $$\text{minimize} \;\; c_i - \mathbf{p}'\mathbf{A}_i \qquad (= \overline{c}_i)$$

  where $\mathbf{p}' = \mathbf{c}_B'\mathbf{B}^{-1}$
  - Find j such that $\overline{c}_j \leq \overline{c}_i$ for all i
- Run revised simplex
  - If $\overline{c}_j \geq 0$, have optimal solution
  - If $\overline{c}_j < 0$, $\mathbf{A}_j$ enters the basis
- Method terminates in the absence of degeneracy

Cutting stock problem

- Fabric rolls of width r
- Sizes of interest $w_1, \ldots, w_m$
  - Example: $r = 10$ and $w_1 = 5$, $w_2 = 4$, $w_3 = 3$.
- Demand b_i for each size w_i
- Minimize the number of rolls needed to satisfy demand

Cutting stock (ctd)

- Each roll is cut according to a certain pattern
- Example: $r = 10$ and $w_1 = 5$, $w_2 = 4$, $w_3 = 3$.
- Allowed patterns:

$$\mathbf{A}_1 = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \qquad \mathbf{A}_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \qquad \mathbf{A}_3 = \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix} \qquad \mathbf{A}_4 = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$$

- A vector

$$\mathbf{A}_j = \begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix}$$

is an allowed pattern if:

$$\sum_{i=1}^{m} a_i w_i \leq r$$
$$a_i \text{ integer}, \quad a_i \geq 0$$

- Let x_j = number of rolls cut according to pattern A_j

$$\begin{aligned} \text{minimize} \quad & \sum_j x_j \\ \text{subject to} \quad & \sum_j A_j x_j = b \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

Cutting stock (ctd)

$$\begin{aligned} \text{minimize} \quad & \sum_j x_j \\ \text{subject to} \quad & \sum_j \mathbf{A}_j x_j = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

- 1. Optimal solution need not be integer
- 2. Number of possible patterns is huge

- 1. Solve LP and round each x_j upwards
- 2. Use delayed column generation

- At each iteration, minimize $\overline{c}_j = 1 - \mathbf{p}'\mathbf{A}_j$
  – maximize $\mathbf{p}'\mathbf{A}_j$

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{m} p_i a_i \\ \text{subject to} \quad & \sum_{i=1}^{m} w_i a_i \leq r \\ & a_i \geq 0, \quad a_i \text{ integer} \end{aligned}$$

- "Knapsack" problem (p_i =value, w_i=weight)
- Despite integrality constraints, can be solved fairly efficiently

Variant with retained columns

- Keep some columns $\mathbf{A}_i$, $i \in I$, in memory
  (The basic columns plus, possibly, more)

- Look for j with $\overline{c}_j < 0$
  - Look only inside the set I
  - Same as solving restricted problem:

$$\begin{array}{ll} \text{minimize} & \mathbf{c}'\mathbf{x} \\ \text{subject to} & \sum_{i \in I} \mathbf{A}_i x_i = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{array}$$

- When at optimal of restricted problem,
  look outside the set I for j with $\overline{c}_j < 0$
- Form new set I (that includes j) and restart

- Extreme variants:
  - $I=$ set of basic indices
  - $I=$ indices of all columns generated in the past
- All variants terminate under nondegeneracy

Cutting plane methods

- Dual of standard form problem:

$$\begin{array}{lll} \text{maximize} & \mathbf{p}'\mathbf{b} & \\ \text{subject to} & \mathbf{p}'\mathbf{A}_i \leq c_i, & i = 1, \ldots, n, \end{array}$$

- Large number n of constraints
- Let $I \subset \{1, \ldots, n\}$
- Solve **relaxed** dual problem

$$\begin{array}{lll} \text{maximize} & \mathbf{p}'\mathbf{b} & \\ \text{subject to} & \mathbf{p}'\mathbf{A}_i \leq c_i, & i \in I, \end{array}$$

- If optimal solution of relaxed problem
  satisfies **all** constraints of original problem,
  then it is optimal for the latter

Cutting planes (continued)

- If optimal solution of relaxed problem is infeasible, bring a violated constraint into I
- Method needs:
  - A way of checking feasibility
  - A way of identifying violated constraints
- One possibility

$$\text{minimize} \;\; c_i - (\mathbf{p}^*)'\mathbf{A}_i$$

- Cutting planes for dual = Column generation for primal
- Options:
  - Retain old constraints
  - Discard (some) inactive constraints