Simplex algorithm (2/2)

15.093: Optimization

Dimitris Bertsimas
Alexandre Jacquillat

Sloan School of Management
Massachusetts Institute of Technology

Reminder: The simplex algorithm

Algorithm

1. *Start with basis* $\boldsymbol{B} = [\boldsymbol{A}_{B(1)}, \cdots, \boldsymbol{A}_{B(m)}]$ *and BFS* $\boldsymbol{x}$.
2. *Compute* $\overline{c}_j = c_j - \boldsymbol{c}_B^\top \boldsymbol{B}^{-1} \boldsymbol{A}_j$
   - *If* $\overline{c}_j \geq 0$ *for all non-basic variables* j, *then* $\boldsymbol{x}$ *is optimal; STOP.*
   - *Else select non-basic variable* j *such that* $\overline{c}_j < 0$; *PROCEED.*
3. *Compute* $\boldsymbol{u} = \boldsymbol{B}^{-1} \boldsymbol{A}_j$.
   - *If* $\boldsymbol{u} \leq \boldsymbol{0}$, *then the cost is unbounded; STOP. Else, PROCEED.*
4. $\theta^* = \min_{i=1,\cdots,m,u_i>0} \dfrac{x_{B(i)}}{u_{B(i)}}$. *Let* l *be such that* $\theta^* = \frac{x_{B(l)}}{u_{B(l)}}$
5. *Form new basis:* $[\boldsymbol{A}_{B(1)} \cdots \boldsymbol{A}_{B(l-1)} \boldsymbol{A}_j \boldsymbol{A}_{B(l+1)} \cdots \boldsymbol{A}_{B(m)}]$
6. *Get new BFS:* $y_j = \theta^*$, $y_{B(i)} = x_{B(i)} - \theta^* u_i$ *for all* i
7. *Go back to Step 2.*

$\rightarrow$ The simplex algorithm proceeds iteratively, moving from one extreme point to an adjacent one until all reduced costs become non-negative

The revised simplex method

Motivation

- Core operations at each simplex iteration
  - Matrix inversion $\boldsymbol{B}^{-1}$: $\mathcal{O}(m^3)$ operations
  - Computing reduced costs $\boldsymbol{c}_B^\top \boldsymbol{B}^{-1}\boldsymbol{A}_j$, $j = 1, \cdots, n$: $\mathcal{O}(mn)$ operations
- Intuition underlying practical simplex implementations:
  - Start with basis $\boldsymbol{B} = [\boldsymbol{A}_{B(1)} \cdots \boldsymbol{A}_{B(m)}]$ and its inverse

$$\boldsymbol{B}^{-1}\boldsymbol{B} = \boldsymbol{I} = [\boldsymbol{e}_1 \ \cdots \ \boldsymbol{e}_m]$$

  - The new basis is “not too far” from the old one:

$$\overline{\boldsymbol{B}} = [\boldsymbol{A}_{B(1)} \cdots \boldsymbol{A}_{B(l-1)} \boldsymbol{A}_j \boldsymbol{A}_{B(l+1)} \cdots \boldsymbol{A}_{B(m)}]$$

$\rightarrow$ The inverse of the new basis should be “not too far” from the old one

$$\boldsymbol{B}^{-1}\overline{\boldsymbol{B}} = [\boldsymbol{e}_1 \ \cdots \ \boldsymbol{e}_{l-1} \ \boldsymbol{u} \ \boldsymbol{e}_{l+1} \ \cdots \ \boldsymbol{e}_m] \quad \text{with } \boldsymbol{u} = \boldsymbol{B}^{-1}\boldsymbol{A}_j$$

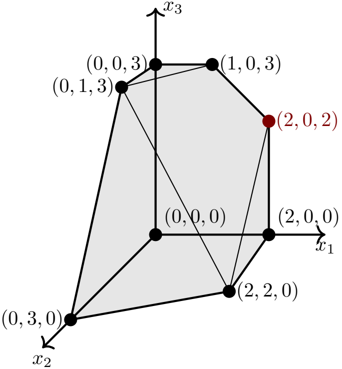
$\rightarrow$ $\overline{\boldsymbol{B}}^{-1}$ by transforming $\boldsymbol{u}$ into $\boldsymbol{e}_l$ through **elementary row operations**, i.e., by adding a multiple of one row to the same or another row

$$\boldsymbol{Q}\boldsymbol{B}^{-1}\overline{\boldsymbol{B}} = \boldsymbol{I} = [\boldsymbol{e}_1 \ \cdots \ \boldsymbol{e}_m] \implies \overline{\boldsymbol{B}}^{-1} = \boldsymbol{Q}\boldsymbol{B}^{-1}$$

Reminder: Example of a simplex iteration

$$\min_{\boldsymbol{x},\boldsymbol{s}\geq\boldsymbol{0}} \quad x_1 + 5x_2 - 2x_3$$

$$\text{s.t.} \quad x_1 + x_2 + x_3 + s_1 = 4$$
$$x_1 + s_2 = 2$$
$$x_3 + s_3 = 3$$
$$3x_2 + x_3 + s_4 = 6$$



$$\boldsymbol{x} = (\mathbf{2}, \mathbf{0}, \mathbf{2}, \mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{4})^\top$$

$$\boldsymbol{B} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \boldsymbol{B}^{-1} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ -1 & 1 & 1 & 0 \\ -1 & 1 & 0 & 1 \end{bmatrix}$$

$$\Longrightarrow \bar{\boldsymbol{c}} = (\mathbf{0}, \mathbf{7}, \mathbf{0}, \mathbf{2}, \mathbf{-3}, \mathbf{0}, \mathbf{0})^\top$$

$$\Longrightarrow \begin{cases} d_5 = \mathbf{1} \\ d_2 = d_4 = \mathbf{0} \\ \boldsymbol{d}_B = -\boldsymbol{B}^{-1}\boldsymbol{A}_5 = (\mathbf{-1}, \mathbf{1}, \mathbf{-1}, \mathbf{-1})^\top \end{cases}$$

$$\Longrightarrow \boldsymbol{y} = (\mathbf{2} - \boldsymbol{\theta}, \mathbf{0}, \mathbf{2} + \boldsymbol{\theta}, \mathbf{0}, \boldsymbol{\theta}, \mathbf{1} - \boldsymbol{\theta}, \mathbf{4} - \boldsymbol{\theta})^\top$$

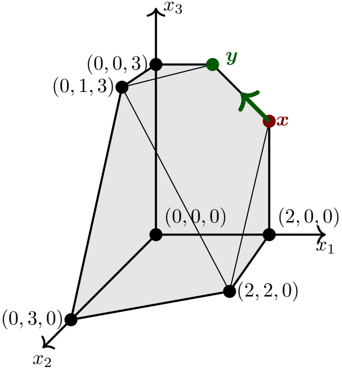
$$\Longrightarrow \theta = 1 \Longrightarrow \boldsymbol{y} = (\mathbf{1}, \mathbf{0}, \mathbf{3}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{3})^\top$$

$$\overline{\boldsymbol{B}} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \ \overline{\boldsymbol{B}}^{-1} = ??$$

Example of the revised simplex method

$$\min_{\boldsymbol{x},\boldsymbol{s}\geq\mathbf{0}} \quad x_1 + 5x_2 - 2x_3$$

$$\text{s.t.} \quad x_1 + x_2 + x_3 + s_1 = 4$$
$$x_1 + s_2 = 2$$
$$x_3 + s_3 = 3$$
$$3x_2 + x_3 + s_4 = 6$$



$$\boldsymbol{B}^{-1} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ -1 & 1 & 1 & 0 \\ -1 & 1 & 0 & 1 \end{bmatrix} \quad \boldsymbol{u} = -\boldsymbol{d}_B = \begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$$

$$\boldsymbol{M} = [\boldsymbol{B}^{-1}\ \boldsymbol{u}] \in \mathbb{R}^{m\times(m+1)}$$

$$\Longrightarrow \boldsymbol{M} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & -1 \\ \underline{-1} & \underline{1} & \underline{1} & \underline{0} & \underline{1} \\ -1 & 1 & 0 & 1 & 1 \end{bmatrix} \begin{array}{l} (a) \\ (b) \\ (c)\ \text{[row } l\text{]} \\ (d) \end{array}$$

$$\Longrightarrow \boldsymbol{M} \leftarrow \begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \underline{-1} & \underline{1} & \underline{1} & \underline{0} & \underline{1} \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix} \begin{array}{l} (a)-(c) \\ (b)+(c) \\ (c) \\ (d)-(c) \end{array}$$

$$\Longrightarrow \overline{\boldsymbol{B}}^{-1} = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

Revised simplex algorithm

Algorithm

1. *Start with basis* $\boldsymbol{B} = [\boldsymbol{A}_{B(1)}, \cdots, \boldsymbol{A}_{B(m)}]$, *BFS* $\boldsymbol{x}$, *and* $\boldsymbol{B}^{-1}$.
2. *Compute* $\bar{c}_j = c_j - \boldsymbol{c}_B^\top \boldsymbol{B}^{-1} \boldsymbol{A}_j$
   - *If* $\bar{c}_j \geq 0$ *for all non-basic variables* j*, then* $\boldsymbol{x}$ *is optimal;* STOP.
   - *Else select non-basic variable* j *such that* $\bar{c}_j < 0$; PROCEED.
3. *Compute* $\boldsymbol{u} = \boldsymbol{B}^{-1} \boldsymbol{A}_j$.
   - *If* $\boldsymbol{u} \leq \boldsymbol{0} \Rightarrow$ *cost unbounded;* STOP. *Else,* PROCEED.
4. $\theta^* = \min\limits_{i=1,\cdots,m, u_i>0} \dfrac{x_{B(i)}}{u_{B(i)}}$. *Let* l *be such that* $\theta^* = \frac{x_{B(l)}}{u_{B(l)}}$
5. *Form new basis:* $[\boldsymbol{A}_{B(1)} \cdots \boldsymbol{A}_{B(l-1)} \boldsymbol{A}_j \boldsymbol{A}_{B(l+1)} \cdots \boldsymbol{A}_{B(m)}]$
6. *Compute* $\overline{\boldsymbol{B}}^{-1}$ *with elementary row operations and go to Step 2.*
   - *Get new BFS:* $y_j = \theta^*$, $y_{B(i)} = x_{B(i)} - \theta^* u_i$ *for all* i
   - $\boldsymbol{M} = [\boldsymbol{B}^{-1}\ \boldsymbol{u}] \in \mathbb{R}^{m \times (m+1)}$; $\begin{cases} \text{row } i \leftarrow \text{row } i - \frac{u_i}{u_l} \times \text{row } l, \forall i \neq l \\ \text{row } l \leftarrow \frac{1}{u_l} \text{row } l \end{cases}$

   $\rightarrow$ *The inverse is in the first* m *columns:* $[\boldsymbol{M}_1 \cdots \boldsymbol{M}_m] = \overline{\boldsymbol{B}}^{-1}$

Revised simplex method: summary

Proposition

The elementary row operations correctly compute $\overline{\boldsymbol{B}}^{-1}$ at each iteration of the revised simplex algorithm

- Reminder on the starting point:

$$\boldsymbol{B}^{-1}\overline{\boldsymbol{B}} = [\boldsymbol{e}_1 \ \cdots \ \boldsymbol{e}_{l-1} \ \boldsymbol{u} \ \boldsymbol{e}_{l+1} \ \cdots \ \boldsymbol{e}_m] \quad \text{with } \boldsymbol{u} = \boldsymbol{B}^{-1}\boldsymbol{A}_j$$

- $\boldsymbol{Q}$ is the matrix associated with elementary row operations:

$$\begin{cases} \text{row } i \leftarrow \text{row } i - \frac{u_i}{u_l} \times \text{row } l, \forall i \neq l \\ \text{row } l \leftarrow \frac{1}{u_l} \text{row } l \end{cases}$$

- By construction: $\boldsymbol{Q}\boldsymbol{u} = \boldsymbol{e}_l$ and $\boldsymbol{Q}\boldsymbol{e}_j = \boldsymbol{e}_j$ for all $j \neq l$

→ $\boldsymbol{Q}\boldsymbol{B}^{-1}\overline{\boldsymbol{B}} = \boldsymbol{I}$ and $\overline{\boldsymbol{B}}^{-1} = \boldsymbol{Q}\boldsymbol{B}^{-1}$

→ Same progress and same convergence as the simplex method, but relies on elementary row operations vs matrix inversion to compute $\boldsymbol{B}^{-1}$

Full tableau implementation

The simplex tableau

- Columns $1, \cdots, n$: $\boldsymbol{B}^{-1}\boldsymbol{A}_1, \cdots \boldsymbol{B}^{-1}\boldsymbol{A}_n$
- Column 0: current solution $\boldsymbol{B}^{-1}\boldsymbol{b} = [x_{B(1)} \ \cdots \ x_{B(m)}]$
- Row 0: reduced costs $\bar{c}_1, \cdots, \bar{c}_n$
- Cell 0: opposite of current cost $-\boldsymbol{c}_B^\top \boldsymbol{x}_B$

$-\boldsymbol{c}_B^\top \boldsymbol{B}^{-1}\boldsymbol{b}$	$\boldsymbol{c}^\top - \boldsymbol{c}_B^\top \boldsymbol{B}^{-1}\boldsymbol{A}$
$\boldsymbol{B}^{-1}\boldsymbol{b}$	$\boldsymbol{B}^{-1}\boldsymbol{A}$

$-\boldsymbol{c}_B^\top \boldsymbol{x}_B$	$\bar{c}_1$	$\cdots$	$\bar{c}_n$
$x_{B(1)}$	$\vert$		$\vert$
$\vdots$	$\boldsymbol{B}^{-1}\boldsymbol{A}_1$	$\cdots$	$\boldsymbol{B}^{-1}\boldsymbol{A}_n$
$x_{B(m)}$	$\vert$		$\vert$

- Example, with three direct variables (x_1, x_2, x_3) and three slack variables (x_4, x_5, x_6), starting with $x_1 = x_2 = x_3 = 0$

		x_1	x_2	x_3	x_4	x_5	x_6
	0	−10	−12	−12	0	0	0
x_4	20	1	2	2	1	0	0
x_5	20	2	1	2	0	1	0
x_6	20	2	2	1	0	0	1

$$\begin{aligned} \min_{\boldsymbol{x} \geq \boldsymbol{0}} \quad & -10x_1 - 12x_2 - 12x_3 \\ \text{s.t.} \quad & x_1 + 2x_2 + 2x_3 \leq 20 \\ & 2x_1 + x_2 + 2x_3 \leq 20 \\ & 2x_1 + 2x_2 + x_3 \leq 20 \end{aligned}$$

Tableau operations

		x_1	x_2	x_3	x_4	x_5	x_6
	0	-10	-12	-12	0	0	0
x_4	20	1	2	2	1	0	0
x_5	20	2	1	2	0	1	0
x_6	20	2	2	1	0	0	1

- Pivot operations to define the tableau with new basis $[\boldsymbol{A}_{B(1)} \cdots \boldsymbol{A}_{B(l-1)} \boldsymbol{A}_j \boldsymbol{A}_{B(l+1)} \cdots \boldsymbol{A}_{B(m)}]$
  - Choose entering column j, with $\bar{c}_j < 0$ (column j: $\boldsymbol{u} = \boldsymbol{B}^{-1}\boldsymbol{A}_j$)
  - Choose exiting column $B(l)$, with $u_{B(l)} > 0$ and smallest ratio $\frac{x_{B(l)}}{u_{B(l)}}$
  - Scale pivot row (l^{th} row) to leave a "1" at the intersection (j^{th} column)
  - Perform pivot operations:

$$\text{new row} \; = \; \text{old row} \; - \; \text{pivot column coefficient} \times \text{new pivot row}$$

→ Application of the elementary row operations:

$$\begin{cases} \text{row } i \leftarrow \text{row } i - \frac{u_i}{u_l} \times \text{row } l, \forall i \neq l \\ \text{row } l \leftarrow \frac{1}{u_l} \text{row } l \end{cases}$$

→ Extension of elementary row operations to objective and reduced costs

Complete example

		x_1	x_2	x_3	x_4	x_5	x_6
	0	−10	−12	−12	0	0	0
x_4	20	1	2	2	1	0	0
x_5	20	2	1	2	0	1	0
x_6	20	2	2	1	0	0	1

		x_1	x_2	x_3	x_4	x_5	x_6
	100	0	−7	−2	0	5	0
x_4	10	0	1.5	1	1	−0.5	0
x_1	10	1	0.5	1	0	0.5	0
x_6	0	0	1	−1	0	−1	1

		x_1	x_2	x_3	x_4	x_5	x_6
	120	0	−4	0	2	4	0
x_3	10	0	1.5	1	1	−0.5	0
x_1	0	1	−1	0	−1	1	0
x_6	10	0	2.5	0	1	−1.5	1

		x_1	x_2	x_3	x_4	x_5	x_6
	136	0	0	0	3.6	1.6	1.6
x_3	4	0	0	1	0.4	0.4	−0.6
x_1	4	1	0	0	−0.6	0.4	0.4
x_2	4	0	1	0	0.4	−0.6	0.4

$$\begin{aligned} \min_{\boldsymbol{x}\geq\mathbf{0}} \quad & -10x_1 - 12x_2 - 12x_3 \\ \text{s.t.} \quad & x_1 + 2x_2 + 2x_3 \leq 20 \\ & 2x_1 + x_2 + 2x_3 \leq 20 \\ & 2x_1 + 2x_2 + x_3 \leq 20 \end{aligned}$$

- Entering variable with negative reduced cost
- Leaving variable, which constrains entering variable the most
- Scale pivot row to leave “1” at intersection
- Update tableau through pivot operations
- Repeat, until all reduced costs are non-negative

Benefits of revised simplex and tableau implementation

- Naive implementation of the simplex method involves computing $\boldsymbol{B}^{-1}$ and the reduced costs of all variables at each iteration
  - $\rightarrow$ $\mathcal{O}(m^3 + mn)$ operations at each iteration
- Revised simplex method proceeds exactly like the simplex method at each iteration but derives $\overline{\boldsymbol{B}}^{-1}$ through elementary row operations
  - $\rightarrow$ Same convergence properties as the simplex method
  - $\rightarrow$ Between $\mathcal{O}(m^2)$ and $\mathcal{O}(mn)$ operations at each iteration
- Full tableau implementation is similar to the revised simplex method, but stores more information than strictly necessary ($\boldsymbol{B}^{-1}\boldsymbol{A}$ vs. $\boldsymbol{B}^{-1}\boldsymbol{u}$)
  - $\rightarrow$ Again, same convergence properties as the simplex method
  - $\rightarrow$ $\mathcal{O}(mn)$ operations at each iteration
  - $\rightarrow$ Interpretable visualization of simplex operations

Full simplex algorithm

How to find an initial solution?

- Easy case: $\mathcal{P} = \{\boldsymbol{x} \in \mathbb{R}^n_+ : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\}$ with $\boldsymbol{b} \geq \boldsymbol{0}$

$$\boldsymbol{A}\boldsymbol{x} + \boldsymbol{s} = \boldsymbol{b},\ \boldsymbol{x}, \boldsymbol{s} \geq \boldsymbol{0} \quad \rightarrow \quad \boldsymbol{x} = \boldsymbol{0},\ \boldsymbol{s} = \boldsymbol{b}$$

- General case in standard form: $\mathcal{P} = \{\boldsymbol{x} \in \mathbb{R}^n_+ : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}\}$
  1. Multiply all rows with a negative b_i coefficient by -1, so $\boldsymbol{b} \geq \boldsymbol{0}$
  2. Introduce auxiliary variables $\boldsymbol{y}$ and find minimum slack:

$$\begin{aligned} \min \quad & y_1 + \cdots + y_m \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{b} \\ & \boldsymbol{x}, \boldsymbol{y} \geq \boldsymbol{0} \end{aligned}$$

  - If the optimal slack is > 0, `STOP`: (LO) is infeasible
  - If the optimal slack is 0 and the basis comprises no auxiliary variable $\boldsymbol{y}$, `STOP`: basic feasible solution for (LO)
  - If the optimal slack is 0 and the basis comprises auxiliary variables $\boldsymbol{y}$, apply the simplex algorithm to eliminate auxiliary variables $\boldsymbol{y}$
    [This is called "Phase I" optimization in the simplex algorithm.]

Example

Linear optimization

$$\min_{\boldsymbol{x}\geq\mathbf{0}} \quad x_1 + x_2 + x_3$$
$$\text{s.t.} \quad x_1 + 2x_2 + 3x_3 = 3$$
$$-x_1 + 2x_2 + 6x_3 = 2$$
$$4x_2 + 9x_3 = 5$$
$$3x_3 + x_4 = 1$$

"Phase I" optimization

$$\min_{\boldsymbol{x}\geq\mathbf{0}} \quad x_5 + x_6 + x_7 + x_8$$
$$\text{s.t.} \quad x_1 + 2x_2 + 3x_3 + x_5 = 3$$
$$-x_1 + 2x_2 + 6x_3 + x_6 = 2$$
$$4x_2 + 9x_3 + x_7 = 5$$
$$3x_3 + x_4 + x_8 = 1$$

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
	-11	0	-8	-21	-1	0	0	0	0
x_5	3	1	2	3	0	1	0	0	0
x_6	2	-1	2	6	0	0	1	0	0
x_7	5	0	4	9	0	0	0	1	0
x_8	1	0	0	3	1	0	0	0	1

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
	-10	0	-8	-18	0	0	0	0	1
x_5	3	1	2	3	0	1	0	0	0
x_6	2	-1	2	6	0	0	1	0	0
x_7	5	0	4	9	0	0	0	1	0
x_4	1	0	0	3	1	0	0	0	1

Example (continued)

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
	-4	0	-8	0	6	0	0	0	7
x_5	2	1	2	0	-1	1	0	0	-1
x_6	0	-1	2	0	-2	0	1	0	-2
x_7	2	0	4	0	-3	0	0	1	-3
x_3	1/3	0	0	1	1/3	0	0	0	1/3

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
	-4	-4	0	0	-2	0	4	0	-1
x_5	2	2	0	0	1	1	-1	0	1
x_2	0	-1/2	1	0	-1	0	1/2	0	-1
x_7	2	2	0	0	1	0	-2	1	1
x_3	1/3	0	0	1	1/3	0	0	0	1/3

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
	0	0	0	0	0	2	2	0	1
x_1	1	1	0	0	1/2	1/2	-1/2	0	1/2
x_2	1/2	0	1	0	-3/4	1/4	1/4	0	-3/4
x_7	0	0	0	0	0	-1	-1	1	0
x_3	1/3	0	0	1	1/3	0	0	0	1/3

$\Longrightarrow$ Initial feasible solution: $x_1 = 1,\ x_2 = 1/2,\ x_3 = 1/3$

$\Longrightarrow$ Initial simplex tableau for the "actual" problem (except reduced costs)

Complete simplex algorithm

$$(LO): \quad \min_{\boldsymbol{x} \in \mathcal{P}} \quad \boldsymbol{c}^\top \boldsymbol{x}, \quad \text{where } \mathcal{P} = \{\boldsymbol{x} \in \mathbb{R}^n_+ : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}\}$$

Algorithm

Phase I

1. *Multiply all rows with a negative b_i coefficient by -1*
2. *Introduce auxiliary variables $\boldsymbol{y}$ and solve:*

$$\min \{y_1 + \cdots + y_m : \boldsymbol{A}\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{b},\ \boldsymbol{x}, \boldsymbol{y} \geq \boldsymbol{0}\}$$

   - ***Detect infeasibility****: If optimal cost is > 0, (LO) is infeasible*
   - *Apply simplex algorithm to eliminate auxiliary variables from basis*

Phase II

1. *Start with basis $\boldsymbol{B} = [\boldsymbol{A}_{B(1)}, \cdots, \boldsymbol{A}_{B(m)}]$ and BFS $\boldsymbol{x}$ from Phase I.*
2. *Compute $\bar{c}_j = c_j - \boldsymbol{c}_B^\top \boldsymbol{B}^{-1} \boldsymbol{A}_j$*
3. *Continue with the simplex algorithm until termination*
   - ***Detect unboundedness****: If $\boldsymbol{u} \leq \boldsymbol{0}$, cost unbounded*
   - ***Detect optimality****: If $\bar{\boldsymbol{c}} \geq \boldsymbol{0}$, $\boldsymbol{x}$ is optimal.*

Advanced topics

Pivoting rules

- Which variable to choose to enter the basis?
  - Choose the first column $\boldsymbol{A}_j$ with negative reduced cost $\bar{c}_j < 0$?
    $\rightarrow$ Little work at each iteration, but progress may be slow
  - Choose a column $\boldsymbol{A}_j$, with the most negative reduced cost $\bar{c}_j < 0$?
    $\rightarrow$ More work at each iteration, but more promising column
  - Choose a column $\boldsymbol{A}_j$ with greatest improvement $\theta^*|\bar{c}_j|$?
    $\rightarrow$ Even more work at each iteration, but convergence in fewer iterations
  - Choose a column $\boldsymbol{A}_j$ with steepest edge $\frac{\boldsymbol{c}^\top(\boldsymbol{y}-\boldsymbol{x})}{\|\boldsymbol{y}-\boldsymbol{x}\|}$?
    $\rightarrow$ Less work than greatest improvement, strong performance in practice
  - Anything else?

- Which variable to choose to exit the basis?
  - Not much flexibility: $\boldsymbol{A}_{B(l)}$ where $l \in \underset{i=1,\cdots,m,u_i>0}{\arg\min} \frac{x_{B(i)}}{u_{B(i)}}$
  - In case of ties, choose the "first" eligible variable
  - Anything else?

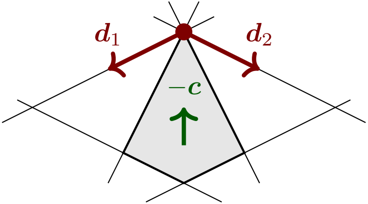
Impact of degeneracy

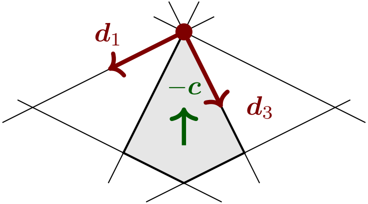
- Recall that simplex terminates in a finite number of iterations if $\mathcal{P}$ admits no degenerate BFS; but what happens with degeneracy?

1. The algorithm may stay in the same solution if it is degenerate: $\boldsymbol{y} = \boldsymbol{x}$
   - Happens if $x_{B(l)} = 0$ and $u_{B(l)} > 0$ for some basic variable l
   - Still, move to new basis $[\boldsymbol{A}_{B(1)} \cdots \boldsymbol{A}_{B(l-1)} \boldsymbol{A}_j \boldsymbol{A}_{B(l+1)} \cdots \boldsymbol{A}_{B(m)}]$
2. The algorithm may lead to a new solution that is degenerate if there is a tie in the smallest ratio rule:

$$\theta^* = \min_{i=1,\cdots,m, u_i>0} \frac{x_{B(i)}}{u_{B(i)}} = \frac{x_{B(l)}}{u_{B(l)}} = \frac{x_{B(l')}}{u_{B(l')}}$$

$\rightarrow$ Can lead to **cycling** in the simplex algorithm





Pivoting rules for anticycling

Definition (lexicographic order)

Let $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^n$. $\boldsymbol{\alpha} \succ_L \boldsymbol{\beta}$ if first non-zero component of $\boldsymbol{\alpha} - \boldsymbol{\beta}$ is positive.

Definition (lexicographic pivoting rule)

1. Choose any $\boldsymbol{A}_j$ with negative reduced cost as entering column.
2. For each i such that $u_i > 0$, divide row i of the tableau by u_i and choose the lexicographically smallest row as the exiting variable $B(l)$

Definition (Bland's rule, or smallest subscript pivoting rule)

1. Define $\boldsymbol{A}_j$ as entering column, with j smallest index such that $\overline{c}_j < 0$.
2. In case of ties, find exiting variables with the smallest subscript.

Theorem

Under the lexicographic pivoting rule and Bland's rule, cycling never occurs and the simplex method terminates after a finite number of iterations.

Conclusion

Summary

Takeaway

Simplex method: invented by George Dantzig in 1947, and widely applied to solve linear optimization problems.

Takeaway

Move from one extreme point to another, maintaining feasibility until optimal solution or direction of unboundedness.

Takeaway

Simple algebra, captured by pivot operations in a tableau.

Takeaway

Finite convergence through careful pivoting rules with degeneracy.

Takeaway

Exponential worst-case performance, but excellent performance in practice.