Ch. 7: Primal simplex algorithm

Standard form:

$$\begin{cases} \min\ c_B^T x_B + c_N^T x_N \\ B x_B + N x_N = b \quad \Big| \, B^{-1} \cdot \\ x_B, x_N \geq 0 \end{cases}$$

$$\rightarrow x_B = B^{-1} b - B^{-1} N x_N, \qquad x_N = 0$$

$$\rightarrow Z = \underbrace{c_B^T \cdot B^{-1}}_{y^T:\ \text{dual variables}} b + \underbrace{\left(c_N^T - c_B^T B^{-1} N\right)}_{\text{slack of dual constraints.}} x_N$$

- dual LP:

$$\begin{cases} \max\ y^T b \\ y^T A \leq C^T \end{cases}$$

For $y^T = c_B^T B^{-1}$:

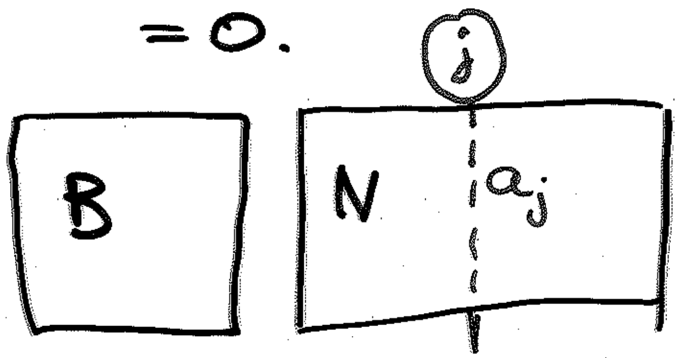
$$y^T B = c_B^T$$

$$\left(c_N^T - y^T N\right) \begin{cases} \geq 0 & \text{optimal} \\ < 0 & \text{opportunity to improve the cost of primal.} \end{cases}$$

Primal simplex (c'ed)

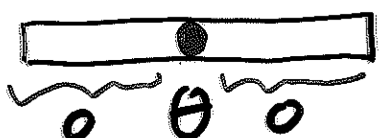
If B is the current basis, we write $z(B)$ for the cost corresponding to B.

$$z(B) = y(B)^T b + \underbrace{(c_N^T - y(B)^T N)\cdot x_N}_{=0.}$$



$\rho_j = c_j - y(B)^T a_j < 0$

If $\theta > 0$ then we hope to reduce $z(B)$ in value. $\leftarrow$ x_N^T:



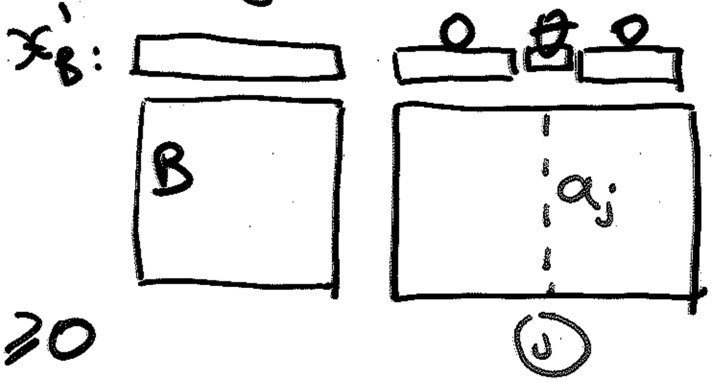
How to choose θ?

$$x'_B = \underbrace{B^{-1}b - B^{-1}a_j\,\theta \geq 0}_{\text{maintain primal feasibility.}}$$

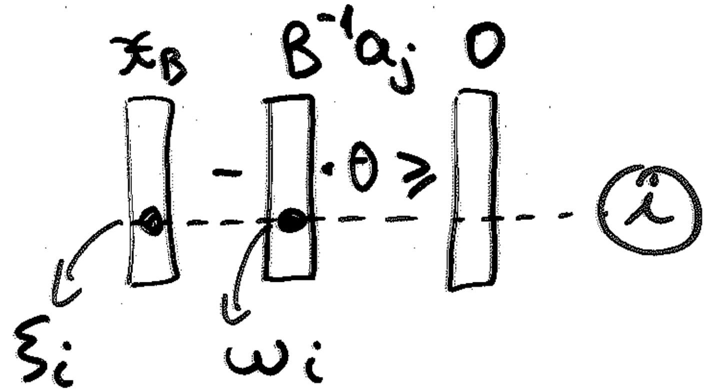
$\downarrow$ new value for variables B.

Primal simplex (c'ed)

Maintain primal feasibility (c'ed):



$$x'_B = x_B - B^{-1} a_j \theta \geq 0$$



$$\theta = \min_{i \in B} \frac{\xi_i}{\omega_i}, \quad \omega_i > 0$$

Q: If $\omega_i \leq 0$ for all $i \in B$?
Then $\theta \to \infty$ maintains feasibility of the primal (problem is unbounded; its solution cost $= -\infty$)

Q: If $\xi_i = 0$ for some $i \in B$?
Degenerate solution, $\theta = 0$; cost of primal is not improved (only basis changes).

Primal simplex (c'ed)

implementation:

a) Find a basic feasible solution:

basis B and $x_B = B^{-1} \cdot b \geq 0$.

b) Recap what we need:

- $x_B = B^{-1} b$: primal
- $z(B) = c_B^T B^{-1} b$: cost
- $y(B) = c_B^T B^{-1}$: dual

$\boxed{\text{implementation 1}}$

- compute B^{-1}
- compute $y(B)$, $\underline{P_j(y(B))}$

OPT | entering variable in B.

- let j = entering variable.
- compute Θ, exit variable i
- $B \leftarrow (B \setminus \{i\}) \cup \{j\}$.
-

Obs: time consuming; compute B^{-1} from scratch while B' not too different from B.

Primal simplex (c'nd)

$\boxed{\text{implementation 2}}$

- use only row matrix operations!

$$\begin{cases} B x_B = b \\ c_B^T x_B + z = 0 \end{cases}, \quad -z = \text{objective}$$

$$\begin{pmatrix} B & 0 \\ c_B^T & 1 \end{pmatrix} \begin{bmatrix} x_B \\ z \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}, \quad z : \text{considered unknown.}$$

solve by computing the inverse of coefficient matrix.

Digression 1 Computing inverse of a matrix Q:

$$(Q \mid I_n) \xrightarrow{\text{row operations}} (I_n \mid Q^{-1})$$

why?

Primal simplex;

$\boxed{\text{implementation 2 (c'ed)}}$

pivot rows do NOT affect column

$$\left(\begin{array}{c|c} B & 0 \\ \hline c_B^T & 1 \end{array}\right)^{-1} = \left(\begin{array}{c|c} Q & 0 \\ \hline q^T & 1 \end{array}\right)$$

, therefore we get in inverse.

Q: What are Q, q^T?

$$\underbrace{\begin{pmatrix} B & 0 \\ c_B^T & 1 \end{pmatrix}\begin{pmatrix} Q & 0 \\ q^T & 1 \end{pmatrix}}_{} = \begin{pmatrix} I_n & 0 \\ 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} B \cdot Q & 0 \\ c_B^T Q + q^T & 1 \end{pmatrix}$$

Therefore $\quad Q = B^{-1}$

$$q^T = -c_B^T B^{-1} = -y(B)^T.$$

nice!

Primal simplex

implementation 2 (c'td)

Step 1 : initialize the tableau & compute its inverse by row operations.

$$\left(\begin{array}{cc|c} B & 0 & b \\ c_B^T & 1 & 0 \end{array}\right)$$

(together with the RHS)

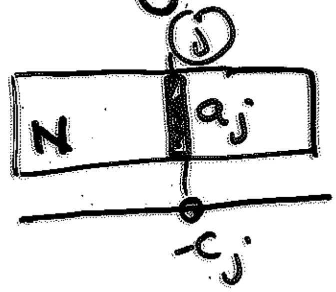
row op. (Gauss-Jordan elim)

inverse:

$$\left(\begin{array}{cc|c} B^{-1} & 0 & x_B \\ -y^T(B) & 1 & -z \end{array}\right)$$

Step 2 : Use $y^T(B)$ to find reduced costs of non-basic variables.

$$\forall j \in N, \quad \rho_j = c_j - y^T(B) \cdot a_j$$



OPT

j : var. entering B.

Primal simplex

$\boxed{\text{implementation 2 (c'ed)}}$

Step 3: j enters the basis, update the tableau.

$$\begin{array}{c} \left(\begin{array}{cc|c|c} B^{-1} & 0 & B^{-1}a_j & x_B \\ -y^T(B) & 1 & \rho_j & -z \end{array}\right) \\ \begin{array}{c} \uparrow \\ \text{entering column } (j) \end{array} \end{array}$$

Recall that

if $B = (b_1\ b_2 \dots b_m)$, then
↓
columns of B

$$B^{-1}a_j = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{pmatrix} \text{ where } a_j = w_1 b_1 + w_2 b_2 + \dots + w_m b_m$$

Step 4: compute the basic var. i leaving the basis:

$$\left(\begin{array}{c|c|c} \dots & B^{-1}a_j & x_B \\ & \rho_j & -z \end{array}\right)$$

$$\theta = \frac{\xi_i}{w_i} = \min_{\ell \in B} \frac{\xi_\ell}{w_\ell}, \quad w_\ell > 0$$

Primal simplex

Implementation 2 (c'ed)

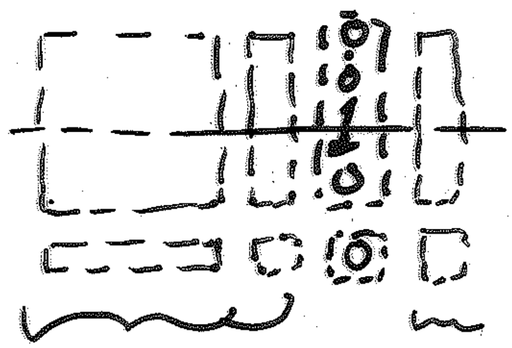
Step 5: Update the inverse of the new tableau:

$$\left(\begin{array}{c|c|c|c} B^{-1} & 0 & B^{-1}a_j & x_B \\ \hline -y^T & 1 & p_j & z \end{array}\right)$$

i : pivot row (step 4)

Perform pivot operation

pivot column



new tableau

Note: One pivot operation is faster than $O(n)$ pivots necessary for calculating the inverse from scratch.

Primal simplex (c'ed)

Digression 2 Updating the matrix inverse if only a column changes:

- Let B be a $m \times m$ non-singular square matrix and B^{-1} its known inverse.

Let $C = \left(B_{1,j-1} \mid a_j \mid B_{j+1,m} \right)$

columns 1 to j-1 of B — new column — columns j+1 to m of B

$$C^{-1} = ?$$

$$B^{-1} \cdot C = \left(B^{-1} \cdot B_{1,j-1} \mid B^{-1} a_j \mid B^{-1} \cdot B_{j+1,m} \right) =$$

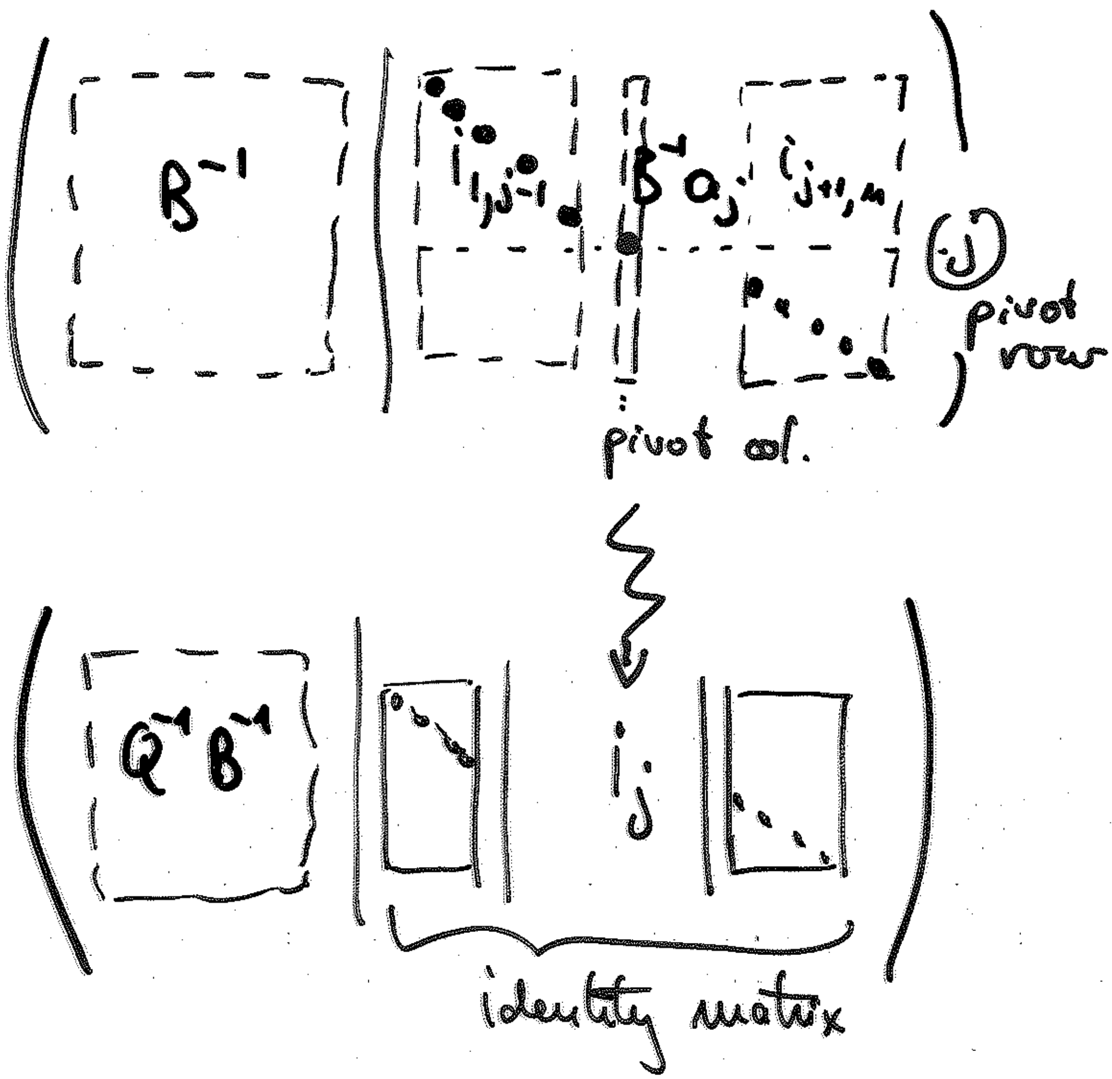
$$= \underbrace{\left(I_{1,j-1} \mid B^{-1} a_j \mid I_{j+1,m} \right)}_{Q \text{ (almost identity)}}$$

$$\Rightarrow C^{-1} = Q^{-1} \cdot B^{-1}$$

Primal-simplex (c'ed)

Digression 2 (c'ed)

Computing $C^{-1} = Q^{-1} \cdot B^{-1}$

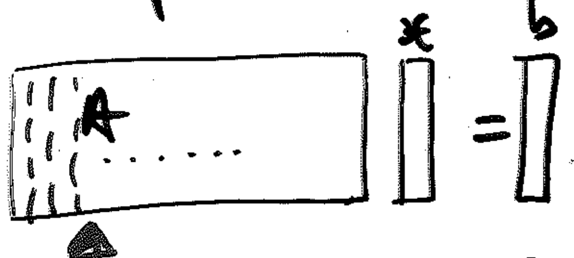


(see digression 1)

This explains Step 5 of primal (revised) simplex algorithm.

Primal simplex (c'ed)

Q: How to find a basic feasible solution to start simplex?

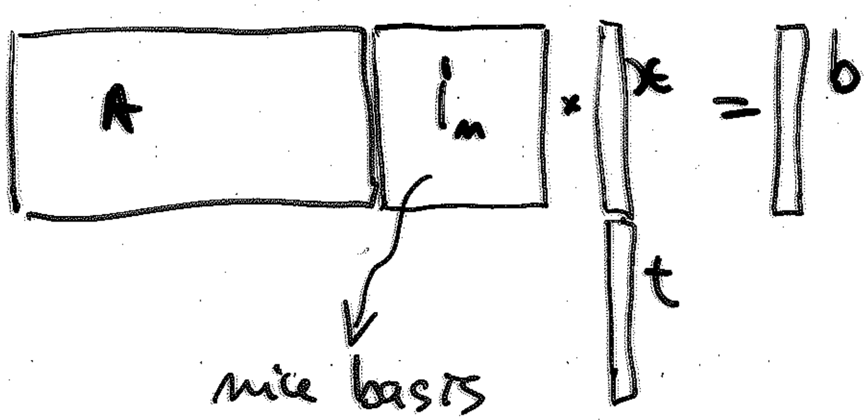


$x \geq 0$

a) Section 3.3 explains how to obtain a basis from the set of column vectors of A

Problem: $x_B \ngeq 0$!

b)



Cost (objective): $\min 1^T t$.

... this is "Phase I" of simplex algorithm...

Primal simplex (c'ed)

Phase I

- Make b positive $(\times(-1))$.
- $$\begin{array}{r|c|cc|c|c}
 & t^T & z & w & & \\
\hline
 & I_m & 0 & 0 & b & a_j \\
\text{cost} \rightarrow & 0 & 1 & 0 & -z & c_j \\
\text{cost phase 1} \rightarrow & 1 & 0 & 1 & -w & 0
\end{array}$$: Build the tableau.

  a_j, c_j, 0: non-basic variables.

  I_m, z, w columns: inverse

$$\begin{pmatrix} I_n & 0 & 0 \\ 0 & 1 & 0 \\ -1^T & 0 & 1 \end{pmatrix}$$

-1^T: duals

- Execute simplex on this tableau.
Objective: eliminate all artificial variables t from basis and obtain objective cost $w = 0$.