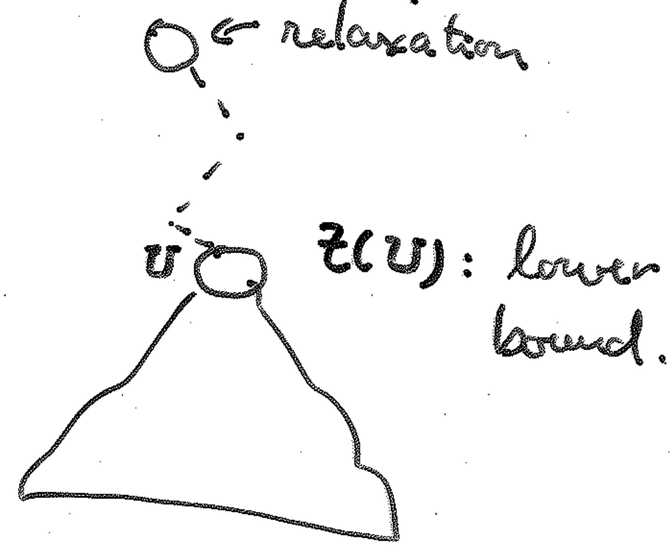
Branch & cut

$$\begin{cases} \min\ c^T x \\ Ax = b \\ x \in \mathbb{Z}_+ \end{cases}$$

B & B (recap)



<u>OBS</u>

B & B faster if

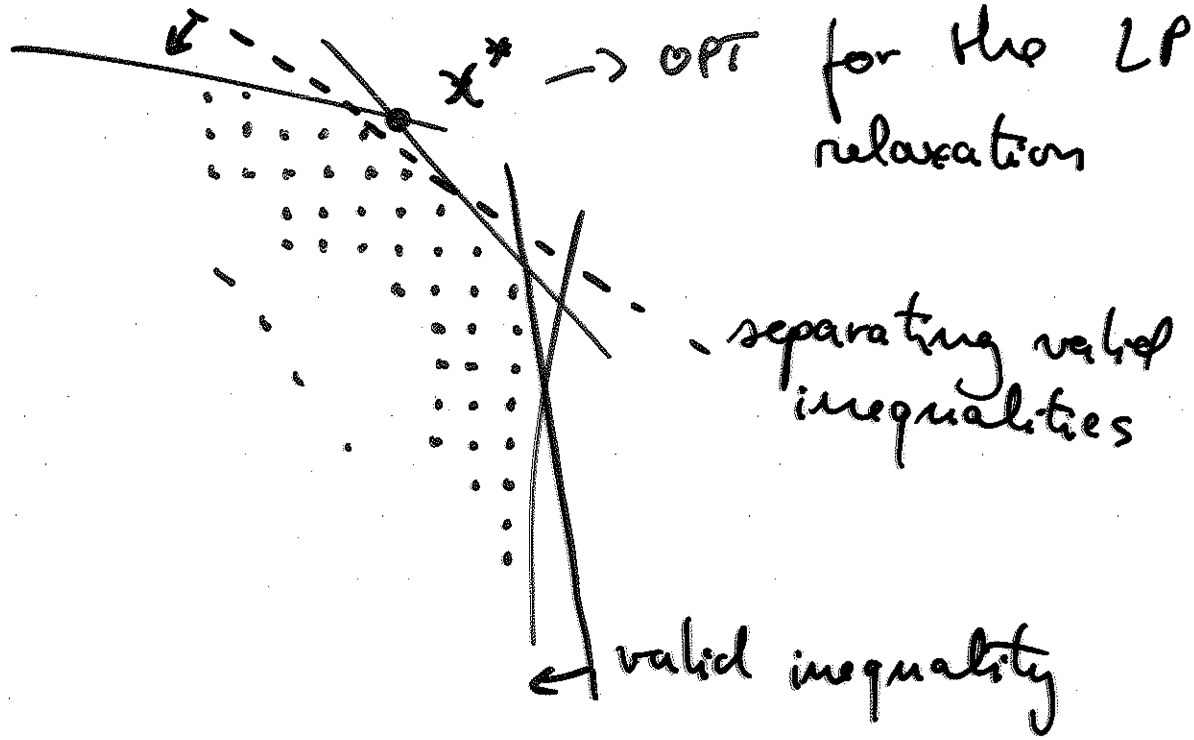
- small upper bound (a good crt. best solution)
- large lower bound (a good relaxation for each problem)

Try to improve the relaxations by adding "valid inequalities"

B & cut.

B&cut (c'ed)

Valid inequalities



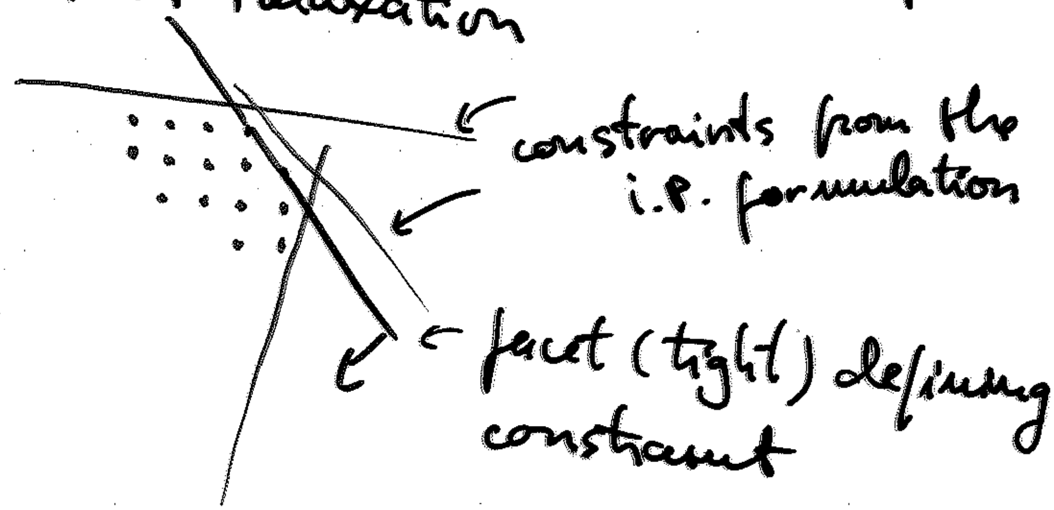
Valid inequality = inequality satisfied by any feasible integer solution to the problem.

Separating valid ineq = the optimal solution to the LP relaxation violates the valid ineq. (lower bound is improved)

B&cut (c'ed)

Valid inequalities

Strong (facet defining) valid inequalities =
= inequalities that prune large portions from the feasible set of the LP relaxation

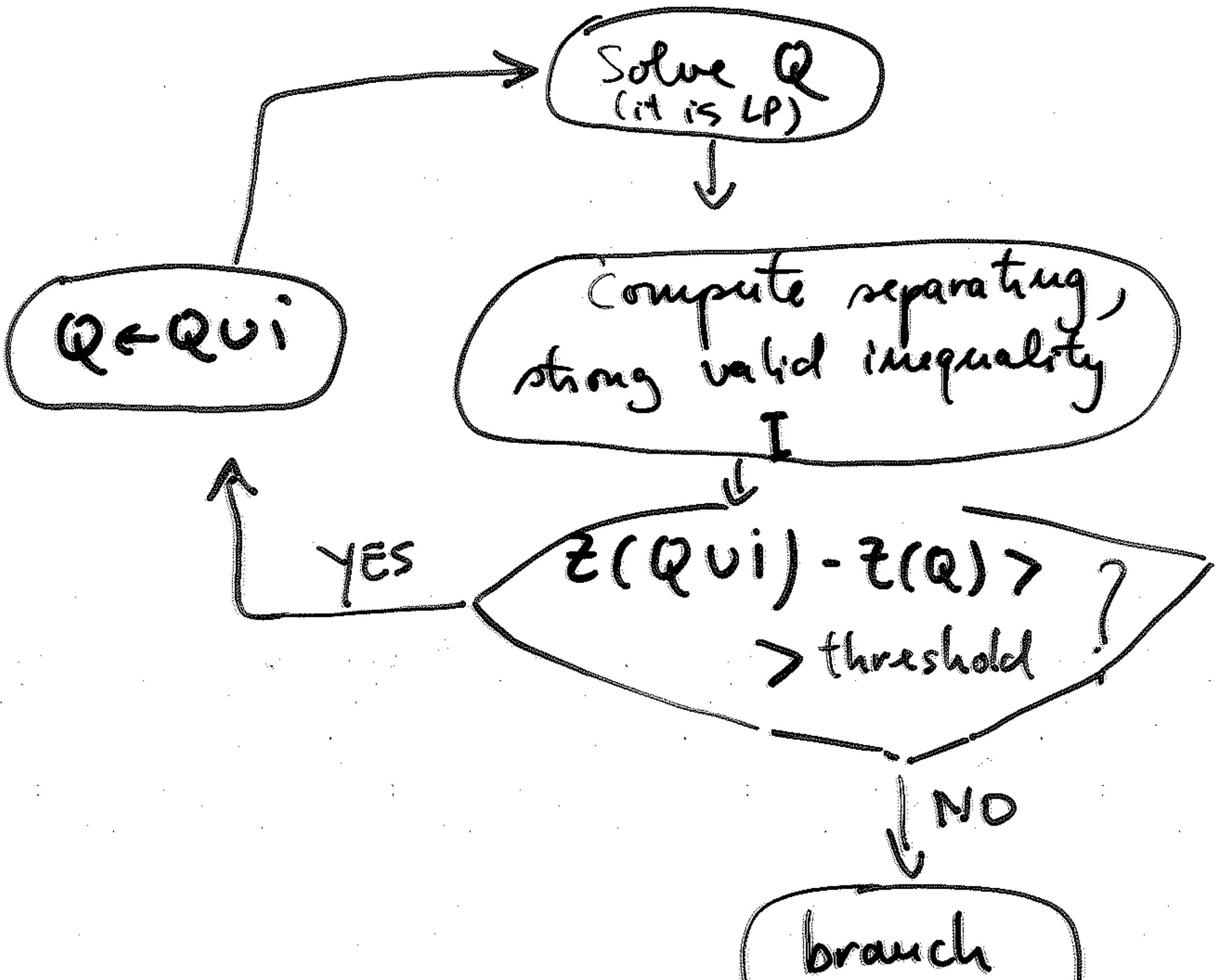


B&cut = B&B + adding strong valid and separating inequalities for each subproblem to improve lower bnd (and thus ~~improve~~ prune larger portions of B&B-tree).

B&cut (c'ed)

Algorithm

- explore B&B tree as in B&B
- when solving subproblem Q :



B&cut

<u>OBS</u> : adding too many inequalities delays solving the LP relaxation (size of problems ↗)

<u>implementation 1</u>

- $Q \leftarrow Q \cup i$
- solve Q using simplex algorithm from scratch.

INEFFICIENT !

<u>implementation 2</u>

- x^* : optimal solution for Q.
- solve $Q \cup i$ starting from x^* and using DUAL SIMPLEX.

B&C (c'ed)

Dual simplex algorithm

- maintain dual feasibility
- maintain complementary slackness
- primal is infeasible. Improve primal feasibility
- all the work done on primal tableau (like normal simplex).

adding I to Q makes x^* infeasible.

However, the dual opt solution y^* remains feasible for $Q \cup I$.

B&cut

$$Q: \begin{cases} \min c^T x \\ Ax = b \\ x \geq 0 \end{cases}$$

B = optimal basis, $x^* = \begin{pmatrix} x_B \\ 0 \end{pmatrix}$

I: $a^T x \leq \beta$ (separating valid inequality)

$$Q_{vi}: \begin{cases} \min c^T x \\ Ax = b \\ a^T x + z = \beta \\ x \geq 0,\ z \geq 0 \end{cases}$$

$\left|\ \begin{matrix} y \\ \pi \end{matrix}\right.$ dual variables

Basis: (infeasible in primal) $\begin{pmatrix} B & 0 \\ a^T & 1 \end{pmatrix}$

Basic variables: x_B, z;

$x_B \geq 0$

$z < 0$

B&cut (c'ed)

QUi (c'ed):

$$\left.\begin{array}{l}\pi = 0 \\ y^T = c_B^T B^{-1} = y^{*T}\end{array}\right\}$$ feasible for QUi because y^* is optimal in Q, thus dual feasible in Q and QUi.

DUAL SIMPLEX:

- B : current basis
- let $r \in B$, $\xi_r < 0$ (infeasible in primal)

Notation: $\bar{b} = B^{-1} b$. thus

$$\xi_r = \bar{b}^{(r)} < 0$$

where $p^{(r)}$: r-th component of vector p.

B&cut (c'ed)

DUAL SIMPLEX (c'ed)

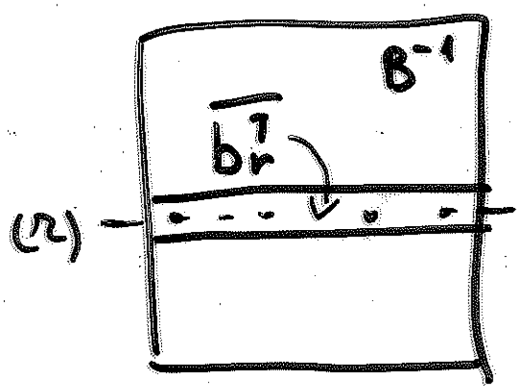
- r leaves the basis, $j \in N$ enters basis.

- $$\frac{\rho_j}{-\overline{a_r^T}(j)} = \min_{\substack{s \in N \\ a_r^T(s) < 0}} \frac{\rho_s}{-\overline{a_r^T}(s)}$$

condition to find the entering variable

where

$$\overline{a_r^T} = \overline{b_r^T}\, A \quad \text{and}$$



, ρ_j : reduced cost of j

why?

- perform usual simplex pivot $B \xleftarrow{} j$, $\rightarrow r$

B & cut (c'ed)

DUAL SIMPLEX (c'ed)

- dual feasibility conditions ?

$$y^T = c_B^T B^{-1} \quad \text{(duals associated with basis } B\text{)}$$

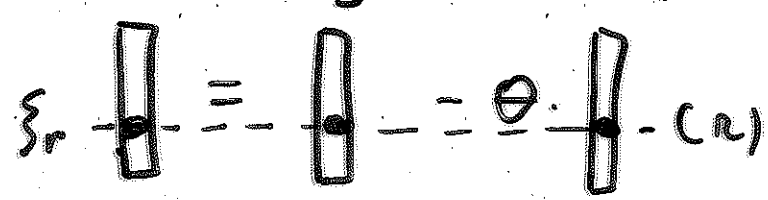
$$c^{(i)} = y^T a_i, \quad i \in B \quad \text{(basic)}$$

$$c^{(i)} \geq y^T a_i, \quad i \in N \quad \text{(non-basic)}$$

- pivoting $\begin{cases} r \in B & \text{leaves} \\ j \in N & \text{enters} \end{cases}$

$$B x_B + \theta a_j = b \quad \Big| \; B^{-1}$$

$$x_B = \underbrace{B^{-1} b}_{\bar{b}} - \theta \underbrace{B^{-1} a_j}_{\bar{a}_j}$$



Goal: $\xi_r = 0$; $\xi_j = \theta \geq 0$

B&cut (c'ed)

- pivoting (c'ed)

$$\xi_j = \theta = \frac{\bar{b}^{(r)}}{\bar{a}_j^{(r)}} \geq 0$$

since $\bar{b}^{(r)} < 0$ (that's why we eliminate r from B)

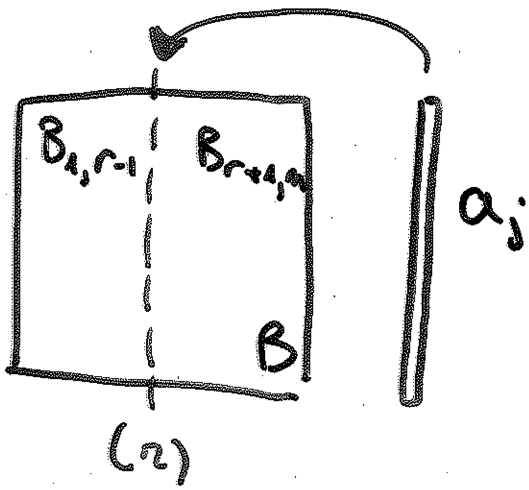
candidates $j \in N$ are only those with $\bar{a}_j^{(r)} < 0$.

This explains condition:

$$\frac{\rho_j}{-\overline{a_r^T}^{(j)}} = \min_{\substack{s \in N \\ \overline{a_r^T}^{(s)} < 0}} \frac{\rho_s}{-\overline{a_r^T}^{(s)}}$$

B & cut (c'ed)

- dual variables during pivoting:



B: old basis

$B_0 = \left(B_{1,r-1} , a_j , B_{r+1,m} \right)$: new basis

What is B_0^{-1} ? $\quad \left(y_0^T = c_{B_0}^T \underline{\underline{B_0^{-1}}} \right)$

⋮

$B_0^{-1} = C^{-1} \cdot B^{-1}$, where

$$C = \begin{pmatrix} I_{r-1} & u & 0 \\ 0^T & \alpha & 0^T \\ 0 & v & I_{m-r-1} \end{pmatrix} \qquad C^{-1} = \begin{pmatrix} I_{r-1} & -\frac{1}{\alpha} u & 0 \\ 0^T & 1/\alpha & 0^T \\ 0 & -\frac{1}{\alpha} \cdot v & I_{m-r-1} \end{pmatrix}$$

$B^{-1} a_j = \bar{a}_j$

B&cut (c'td)

- dual variables (c'td)

$$B_0 = \begin{pmatrix} B_{1,r-1} & a_j & B_{r+1,m} \end{pmatrix}$$

$$c_{B_0}^T = \Big(c_{B_1}^T ,\ \underset{\substack{\uparrow \\ \text{new coefficient}}}{\underline{c^{(j)}}} ,\ c_{B_2}^T \Big)$$

$$y_0^T = \underbrace{c_{B_0}^T C^{-1}}\, B^{-1}$$

$$\begin{pmatrix} c_{B_1}^T & c^{(j)} & c_{B_2}^T \end{pmatrix} \times \begin{pmatrix} I_{r-1} & -\frac{1}{\alpha} u & 0 \\ 0^T & \frac{1}{\alpha} & 0^T \\ 0 & -\frac{1}{\alpha} v & I_{m-r} \end{pmatrix} =$$

$$= \underbrace{\left(c_{B_1}^T ,\ \underbrace{\left(-\frac{1}{\alpha} \left(c_{B_1}^T u - c^{(j)} + c_{B_2}^T v \right) \right)}_{s} ,\ c_{B_2}^T \right)}_{\text{row vector}}$$

B&cut (c'ed)

- dual variables (c'ed)

$$y_0^T = \left(c_{B_1}^T ,\ c^{(r)} + \left(-c^{(r)} + s\right),\ c_{B_2}^T \right) \times B^{-1} =$$

$$= \underbrace{y^T}_{\text{previous dual}} + \underbrace{\left(-c^{(r)} + s\right)}_{\text{scalar value}} \cdot \underbrace{\overline{b_r^T}}_{\text{r'th row of } B^{-1}}$$

$$s - c^{(r)} = -\frac{1}{\alpha}\left(c_{B_1}^T u + c^{(r)}\cdot\alpha + c_{B_2}^T v - \right.$$

$$\left. - c^{(r)}\cdot\alpha - c^{(j)} \right) - c^{(r)} =$$

$$= -\frac{1}{\alpha}\left(c_B^T B^{-1} a_j - \cancel{c^{(r)}\cdot\alpha} - c^{(j)} \right) \cancel{- c^{(r)}}$$

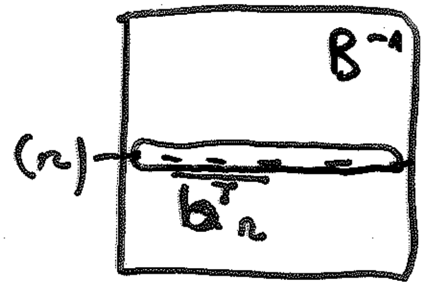
$$\left(\text{recall } \begin{pmatrix} u \\ \alpha \\ v \end{pmatrix} = B^{-1} a_j \right)$$

$$\boxed{s - c^{(r)} = -\frac{1}{\alpha}\left(y^T a_j \right) + \frac{c^{(j)}}{\alpha} = \frac{\rho_j}{\alpha}}$$

B& cut (c'ed)

- dual variables (c'ed)

$$y_0^T = y^T + \frac{\rho_j}{\alpha}\,\overline{b}_r^T$$



pivot $B \genfrac{}{}{0pt}{}{\leftarrow j}{\rightarrow r}$

$$\alpha = \left(B^{-1}a_j\right)^{(r)} = \overline{b}_r^T \cdot a_j$$

ρ_j : old reduced cost

- dual feasibility:

$\forall h \in N$ (non basic)

$$\rho_{0h} = c^{(h)} - y_0^T a_h = c^{(h)} - y^T a_h - \frac{\rho_j}{\alpha}\,\overline{b}_r^T \cdot a_h$$

(new reduced cost)

$$= \rho_h - \rho_j \cdot \frac{\overline{b}_r^T \cdot a_h}{\overline{b}_r^T \cdot a_j} \geq 0$$

B&B cut (c'td)

- dual feasibility conditions:

Using the notation from p. 9 (dual simplex algorithm):

$h \leftrightarrow s$ (replace indices, sorry)

$$\overline{b_r^T} \cdot a_s = \overline{a_r^T}^{(s)} \qquad \left(\overline{a_r^T} = \overline{b_r^T} \cdot A\right)$$

$$\overline{b_r^T}\, a_j = \overline{a_r^T}^{(j)}$$

since $\overline{a_r^T}^{(s)} < 0$ (to make progress)

$$\Rightarrow \frac{\rho_s}{-\overline{a_r^T}^{(s)}} \geqslant \frac{\rho_j}{-\overline{a_r^T}^{(j)}}$$

which explains the test for entering variable in dual simplex (p. 9).