interior point algorithms for linear programming

Linear programming :

- simplex
  - efficient in practice
  - exponential complexity in worst case

- interior point algorithms
  - efficient in practice on large scale problems
  - polynomial complexity in worst case

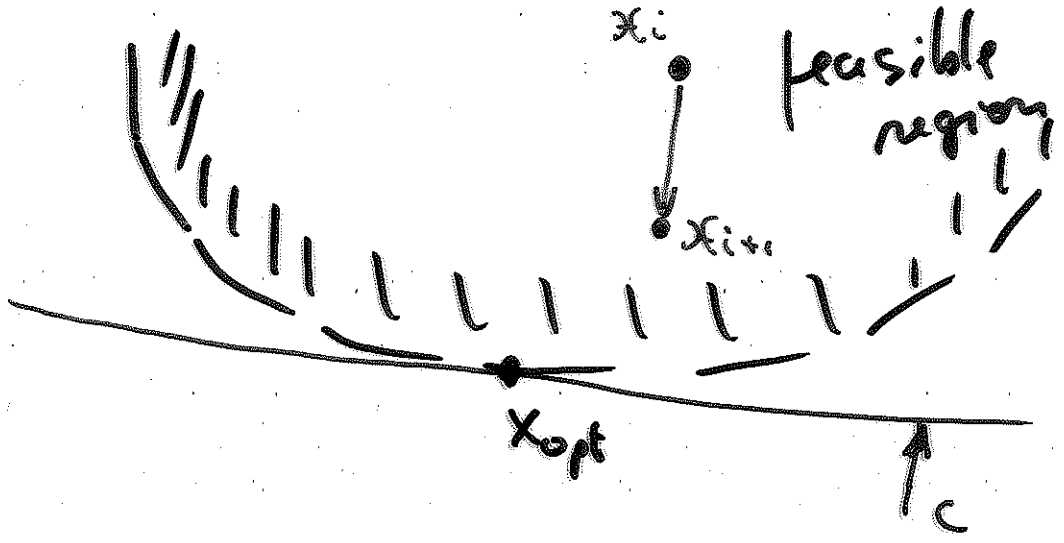
Karmarkar (1984) : first to show that interior point algorithm works reasonably well in practice for LP
(Khachiyan gave the first polynomial time in worst case algorithm to solve LP - 1984).

interior point (cont'd)

Affine scaling algorithm:

ideas:

1) Move through the interior of the feasible region towards the optimal pt.
2) Move in the direction of fastest improvement of the objective
3) Transform the feasible region (affine scaling) so that crt. solution is in the center of the feasible region.



interior-point (cont'd)

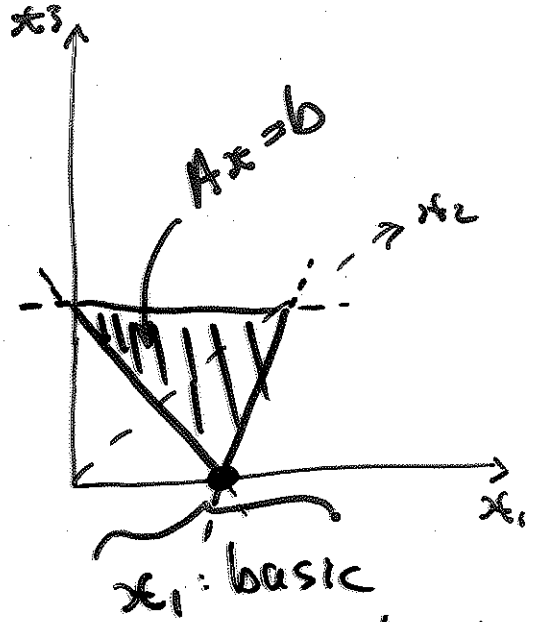
$$\min c^T x$$
$$Ax = b$$
$$x \geq 0$$

3 variables
1 constraint

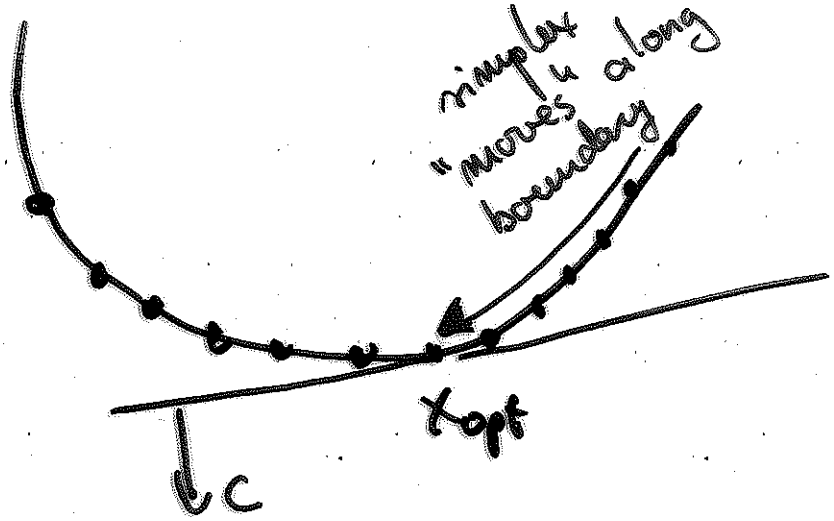
a) Simplex:

$$\begin{cases} x_B = B^{-1}b \geq 0 \\ x_N = 0 \end{cases}$$

Each vertex on the boundary seen by simplex = pivoting.
There are exponentially many vertices in worst case

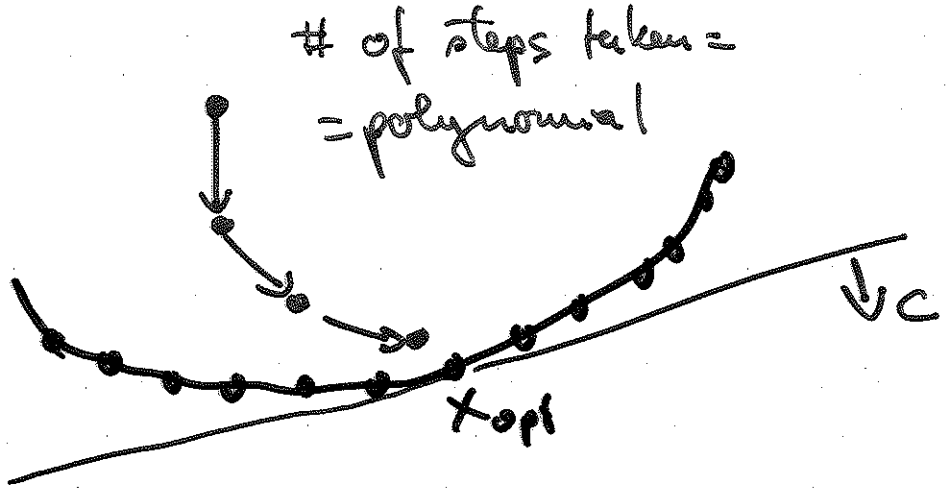


x_1 : basic
x_2, x_3 : non-basic



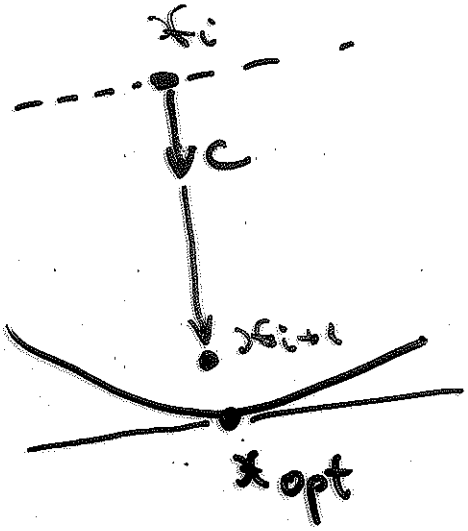
interior point (cont'd)

b) interior pt algorithm



i.pt moves in the interior of the feasible region towards x_{opt}, getting very close to x_{opt} but not quite attaining it.

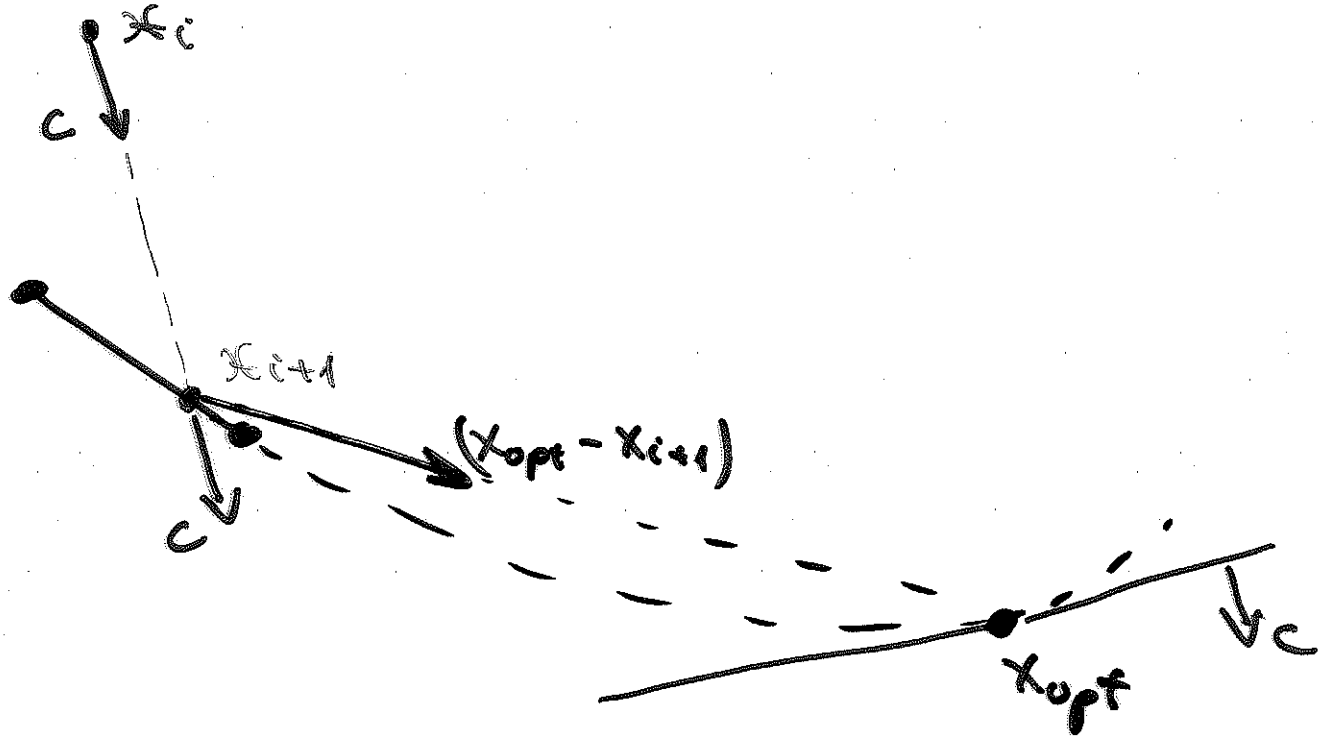
How?



- move in the direction c of steepest descent.
- if x_i is in the "middle" of feasible region, chances are your next solution x_{i+1} is very close to x_{opt} (like in figure)

interior pt (c'ed)

Q: Why not move all the way to the boundary?



At x_{i+1} moving along c to improve the cost will give an infeasible solution. x_{opt} is unknown, so we cannot move towards it, unless we move along the boundary (simplex)

Interior pt (cont'd)

Affine scaling algorithm:

$$\begin{cases} \max z = x_1 + 2x_2 \\ x_1 + x_2 + x_3 = 8 \\ x_i \geq 0 \end{cases}$$

$\left(X_{opt}^T = (0 \;\; 8 \;\; 0);\; z_{opt} = 16 \right)$

- $x_1^T = (2 \;\; 2 \;\; 4)$

Direction of movement?

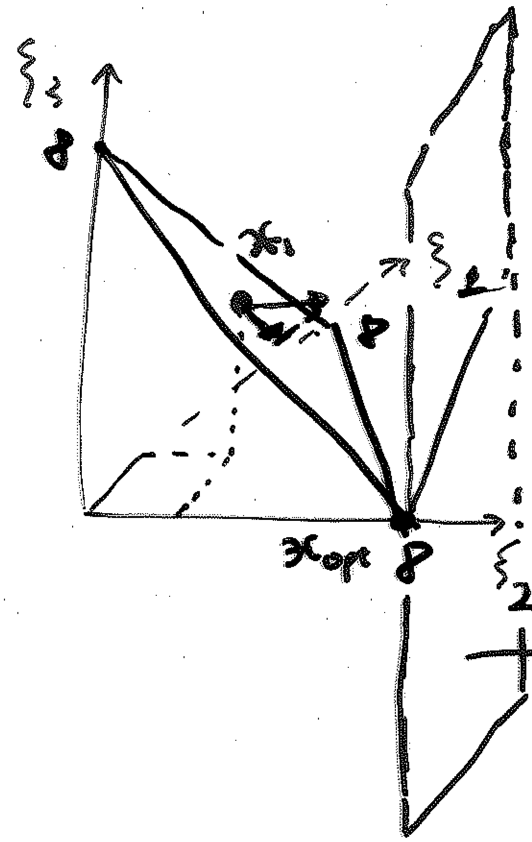
$c^T = (1 \;\; 2 \;\; 0)$

Ex $x_2^T = x_1^T + c^T = (3 \;\; 4 \;\; 4)$

infeasible

→ drop a $\perp$ to the subspace from the infeasible point. The projection becomes the direction of improvement.

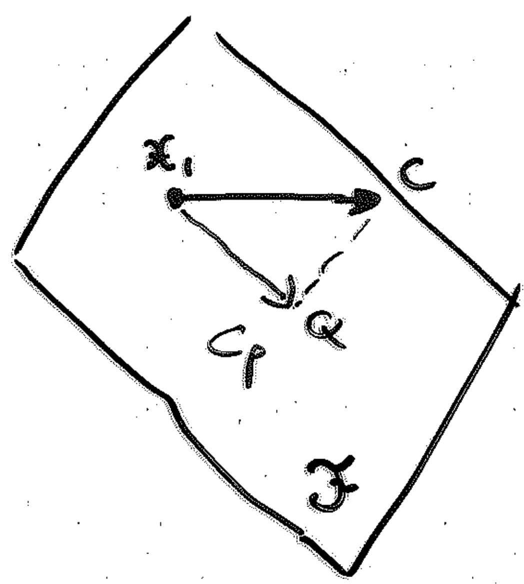
interior pt (c'ed)



$\rightarrow c^T = (1\ 2\ 0)$

Vector ⊥ to the feasible region subspace:

$p = (1\ 1\ 1)$



$x_1 + c - \theta \cdot p \in \mathcal{F}$

$(3\ 4\ 4) - \theta(1\ 1\ 1) \in \mathcal{F}$

$3 - \theta + 4 - \theta + 4 - \theta = 8$

$\Rightarrow \theta = 1$

Point Q corresponds to vector $(2\ 3\ 3)$

But

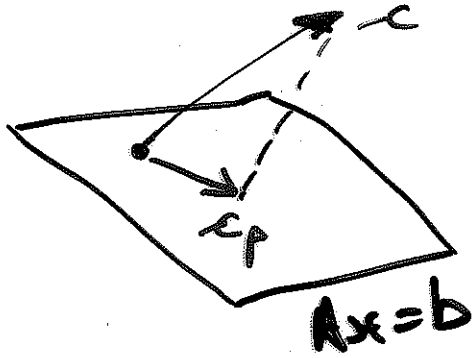
$(2\ 3\ 3) = x_1 + c_p$

$\Rightarrow c_p = (2\ 3\ 3) - (2\ 2\ 4) =$

$= (0\ 1\ -1)$

interior pt (c'ed)

Projection, general case:



$c_p = P \cdot c$ where

$P = I_n - A^T (A A^T)^{-1} A$

Q: How far can we go along c_p?

$x_{i+1} = x_i + \theta c_p \geq 0.$

$(2\ 2\ 4) + \theta (0\ 1\ -1) \geq 0$

Like for simplex, let

$$\theta = \min_{\substack{j \in \{1..m\} \\ c_p(j) < 0}} \frac{\xi_j}{-c_p(j)}$$

where $x_i^T = (\xi_1 \dots \xi_j \dots \xi_m)$.

Use: $x_{i+1} = x_i + \alpha \theta c_p$, $\alpha < 1$ (for ex 0.9)

interior pt (c'ed)

Affine scaling:

$x_1^T = (2 \; 2 \; 4)$

The constraints where we would get stuck if we are too close are

$$\boxed{x \geq 0}$$

There are many notions of "centre" of feasible region. Affine scaling = = change (scaling) of variables so that the current solution becomes

$$\tilde{x}^T = (1 \; 1 \; \ldots \; 1)$$

CY $\quad \tilde{x}_1^T = \underbrace{\begin{pmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{4} \end{pmatrix}}_{D_1^{-1}} x_1$ where

$$D_1 = \text{diag}(x_1) = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{pmatrix}$$

interior pt (c'ed)

Affine scaling (transformed problem):

$$(A D_1)(D_1^{-1} x_1) = b$$

$$\tilde{A} \quad \tilde{x}_1$$

$$\max \; -c^T D_1 \, D_1^{-1} x$$

$$\tilde{c}^T \quad \tilde{x}$$

Interior pt (cont'd)

Karmakar's algorithm (affine scaling):

- Let x_i be crt solution,

$$D = \text{diag}(x_i),$$

$$\tilde{A} = A \cdot D \quad , \quad \tilde{c}^T = c^T D .$$

- Compute projection matrix

$$P = I_n - \tilde{A}^T (\tilde{A}\tilde{A}^T)^{-1} \tilde{A} .$$

and gradient

$$c_p = -P\tilde{c} \quad \text{(for minimization prob.)}$$

- Compute θ so that

$$1 + \theta c_p \geq 0$$

- Compute the next solution:

$$x_{i+1} = D \cdot (1 + \alpha \theta c_p)$$

- Repeat until stopping condition (ex: very little improvement on cost).