NTNU | Norwegian University of Science and Technology

TTK4135 – Lecture 5
Solving LPs – the Simplex method

Lecturer: Lars Imsland

Purpose of lecture

- Brief recap previous lecture
- The geometry of the feasible set
- Basic feasible points, “The fundamental theorem of linear programming”
- **The simplex method**
- Example 13.1
- Some implementation issues

Reference: N&W Ch.13.2-13.3, also 13.4-13.5

Linear programming, standard form and KKT: recap

LP: $\min_{x \in \mathbb{R}^n} c^T x$ subject to $\begin{cases} a_i x = b_i, & i \in \mathcal{E} \\ a_i x \geq b_i, & i \in \mathcal{I} \end{cases}$

LP, standard form: $\min_{x \in \mathbb{R}^n} c^T x$ subject to $\begin{cases} Ax = b \\ x \geq 0 \end{cases}$

Lagrangian: $\mathcal{L}(x, \lambda, s) = c^T x - \lambda^T (Ax - b) - s^T x$

KKT-conditions (LPs: necessary *and* sufficient for optimality):

$$
\begin{aligned}
A^T \lambda^* + s^* &= c, \\
Ax^* &= b, \\
x^* &\geq 0, \\
s^* &\geq 0, \\
x_i^* s_i^* &= 0, \quad i = 1, 2, \ldots, n
\end{aligned}
$$

Duality

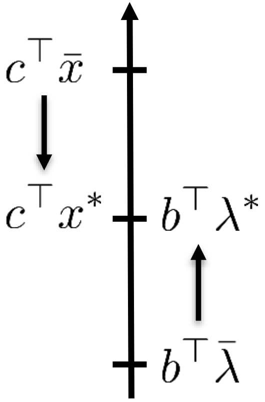
Primal problem

$$\begin{aligned} \min_{x} \quad & c^\top x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

Dual problem

$$\begin{aligned} \max_{\lambda, s} \quad & b^\top \lambda \\ \text{s.t.} \quad & A^\top \lambda + s = c \\ & s \geq 0 \end{aligned}$$

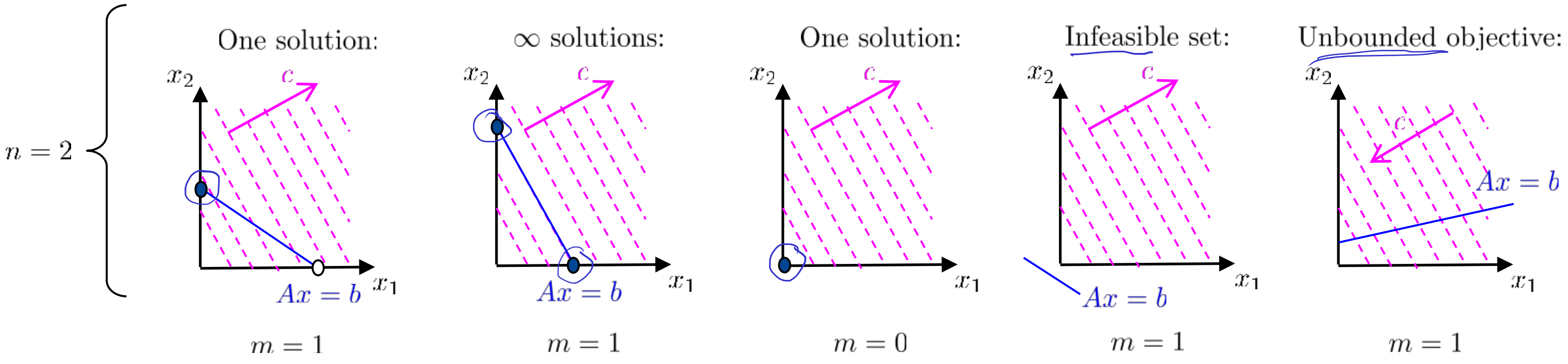
- Primal and dual have same KKT conditions!
- Equal optimal value: $c^\top x^* = b^\top \lambda^*$
- Weak duality: $c^\top \bar{x} \geq b^\top \bar{\lambda}$ ($\bar{x}$, $\bar{\lambda}$ feasible)
- Duality gap: $c^\top \bar{x} - b^\top \bar{\lambda}$
- Strong duality (Thm 13.1):
  - i) If primal or dual has finite solution, both are equal
  - ii) If primal or dual is unbounded, the other is infeasible

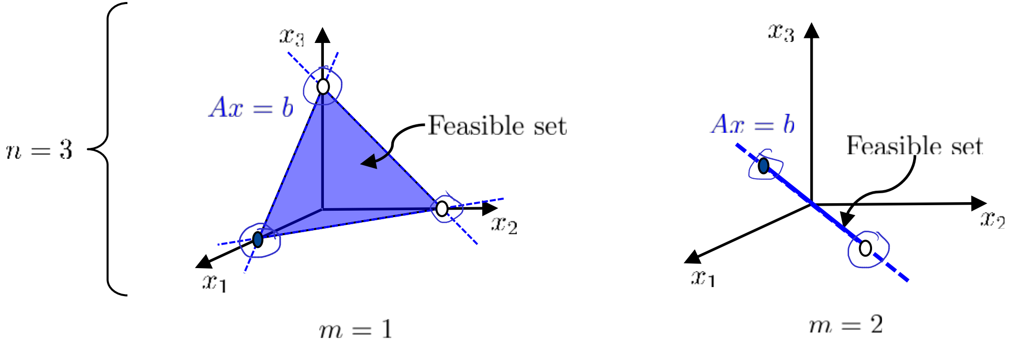


$$c^\top \bar{x} \geq c^\top x^* = b^\top \lambda^* \geq b^\top \bar{\lambda}$$

LP: Geometry of the feasible set

$$\min_x \quad c^\top x \quad \text{s.t.} \quad Ax = b, \quad x \geq 0$$





- ● Basic optimal point (BOP)
- ○ Basic feasible point (BFP) (if they exist)

In general, the BFP has at most *m* non-zero components

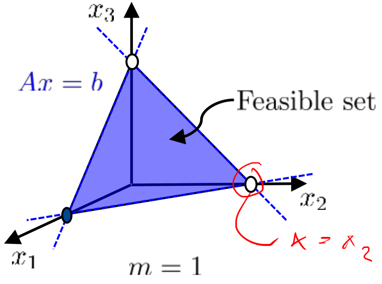
Basic feasible point (BFP)

A point x is a basic feasible point if

- x is feasible
- There is an index set $\mathcal{B}(x) \subset \{1, \ldots, n\}$ such that
  - $\mathcal{B}(x)$ contains *m* indices
  - $i \notin \mathcal{B}(x) \quad \Rightarrow \quad x_i = 0$
  - $B = [A_i]_{i \in \mathcal{B}(x)}$ is non-singular, $B \in \mathbb{R}^{m \times m}$

  ↳ Always holds if A is full row rank

- $\mathcal{B}(x)$ is called a basis for the LP
- The indices not in $\mathcal{B}(x)$ are called $\mathcal{N}(x)$



$$\min_x \quad c^\top x$$
$$\text{s.t.} \quad Ax = b$$
$$x \geq 0$$

$[A_1 \ A_2 \ A_3]$

$\mathcal{B}(x) = \{2\}, \ \mathcal{N}(x) = \{1,3\}$

$B = A_2$ (scalar)

Ex. $n = 5, \ m = 2, \ \mathcal{B}(x) = \{2,3\}$

$$Ax = b$$

$$\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \begin{bmatrix} 0 \\ x \\ x \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} * \\ * \end{bmatrix}$$

B

$\to 0$

$$\mathcal{N}(x) = \{1, 2, \ldots, n\} \setminus \mathcal{B}(x)$$
$$= \{1, 4, 5\}$$

Facts about Simplex method

$$\begin{aligned} \min_x \quad & c^\top x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

The Simplex method generates iterates that are BFP, and converge to a solution if

1) there are BFPs, and
2) one of them is a solution (Basic Optimal Point, BOP)

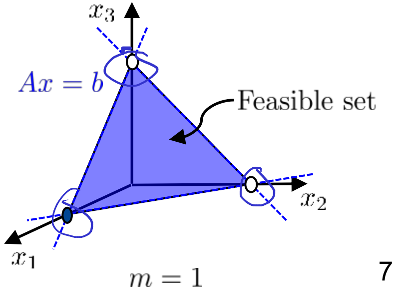
Theorem 13.2 (Fundamental theorem of Linear Programming): For standard form LP

1) If there is a feasible point, there is a BFP
2) If the LP has a solution, one solution is a BOP
3) If LP is feasible and bounded, there is a solution

Theorem 13.3: All vertices of the feasible polytope

$$\{x \mid Ax = b,\ x \geq 0\}$$

are BFPs (and all BFPs are vertices)



Facts about Simplex method, cont'd

Degeneracy: A LP is *degenerate* if there is a BFP x with $x_i = 0$ for some $i \in \mathcal{B}(x)$

Theorem 13.4: If an LP is bounded and non-degenerate, the Simplex method terminates at a BOP

LP KKT conditions (necessary&sufficient)

$$\begin{aligned} \min_x \quad & c^\top x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

- Simplex method iterates BFPs until one that fulfills KKT is found.

$$\begin{aligned}
A^T\lambda + s &= c, && \text{(KKT-1)} \\
Ax &= b, && \text{(KKT-2)} \\
x &\geq 0, && \text{(KKT-3)} \\
s &\geq 0, && \text{(KKT-4)} \\
x_i s_i &= 0, \quad i = 1, 2, \ldots, n && \text{(KKT-5)}
\end{aligned}$$

- Each step is a move from a vertex to a neighboring vertex *(one change in the basis),* that decreases the objective

Simplex definitions

$\mathcal{B}(x) = \{2, 3\}$

$$\begin{pmatrix} A_{11} & \boxed{\begin{matrix} A_{12} & A_{13} \\ A_{22} & A_{23} \end{matrix}} & A_{14} & A_{15} \\ A_{21} & & A_{24} & A_{25} \end{pmatrix} \begin{pmatrix} 0 \\ * \\ * \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

B $\quad x_B$

$$N = \begin{bmatrix} A_{11} & A_{14} & A_{15} \\ A_{21} & A_{24} & A_{25} \end{bmatrix}$$

$x \to \mathcal{B}(x)$ $\quad \mathcal{N}(x) = \{1, \dots, n\} \setminus \mathcal{B}(x)$

$B = [A_i]_{i \in \mathcal{B}(x)}$ $\quad N = [A_i]_{i \in \mathcal{N}(x)}$

$x_B = [x_i]_{i \in \mathcal{B}(x)}$ $\quad x_N = [x_i]_{i \in \mathcal{N}(x)} = 0$

$s_B = [s_i]_{i \in \mathcal{B}(x)}$ $\quad s_N = [s_i]_{i \in \mathcal{N}(x)}$

$c_B = [c_i]_{i \in \mathcal{B}(x)}$ $\quad c_N = [c_i]_{i \in \mathcal{N}(x)}$

One step of Simplex-algorithm

$$
\begin{aligned}
A^T\lambda + s &= c, && \text{(KKT-1)}\\
\rightarrow\ Ax &= b, && \text{(KKT-2)}\\
x &\ge 0, && \text{(KKT-3)}\\
s &\ge 0, && \text{(KKT-4)}\\
x_i s_i &= 0, \quad i = 1,\dots,n && \text{(KKT-5)}
\end{aligned}
$$

Given x, a BFP:

KKT-2: $Ax = Bx_B + \overbrace{Nx_N}^{=0} = b$ (ok)

KKT-3: $x_B = B^{-1}b \ge 0$, $x_N = 0$ (ok)

KKT-5: $x^Ts = x_B^T s_B + \underbrace{x_N^T s_N}_{=0} = 0$. Set $s_B = 0$. (ok)

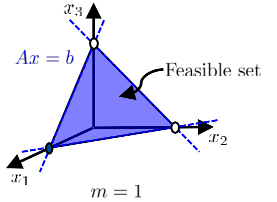
KKT-1:

$$
\begin{bmatrix} \overset{=0}{s_B} \\ s_N \end{bmatrix} + \begin{bmatrix} B^T \\ N^T \end{bmatrix}\lambda = \begin{bmatrix} c_B \\ c_N \end{bmatrix} \Rightarrow \begin{cases} \lambda = (B^T)^{-1} c_B \\ s_N = c_N - N^T\lambda \end{cases} \text{(ok)}
$$

$s_N \ge 0$?

KKT-4: Ok if $s_N \ge 0$!

The BFP x is a BOP, a solution!



One step of Simplex-algorithm

$$
\begin{aligned}
A^T\lambda + s &= c, && \text{(KKT-1)}\\
Ax &= b, && \text{(KKT-2)}\\
x &\geq 0, && \text{(KKT-3)}\\
s &\geq 0, && \text{(KKT-4)}\\
x_i s_i &= 0, \quad i = 1, \dots, n && \text{(KKT-5)}
\end{aligned}
$$

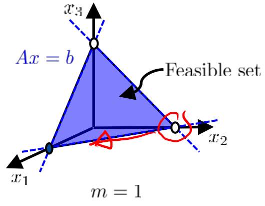
If $s_N \ngeq 0$: (Find new BFP)

- Choose one index $q \in \mathcal{N}$ s.t. $s_q < 0$
- Increase x_q along $Ax = b$ until a componenent becomes zero.

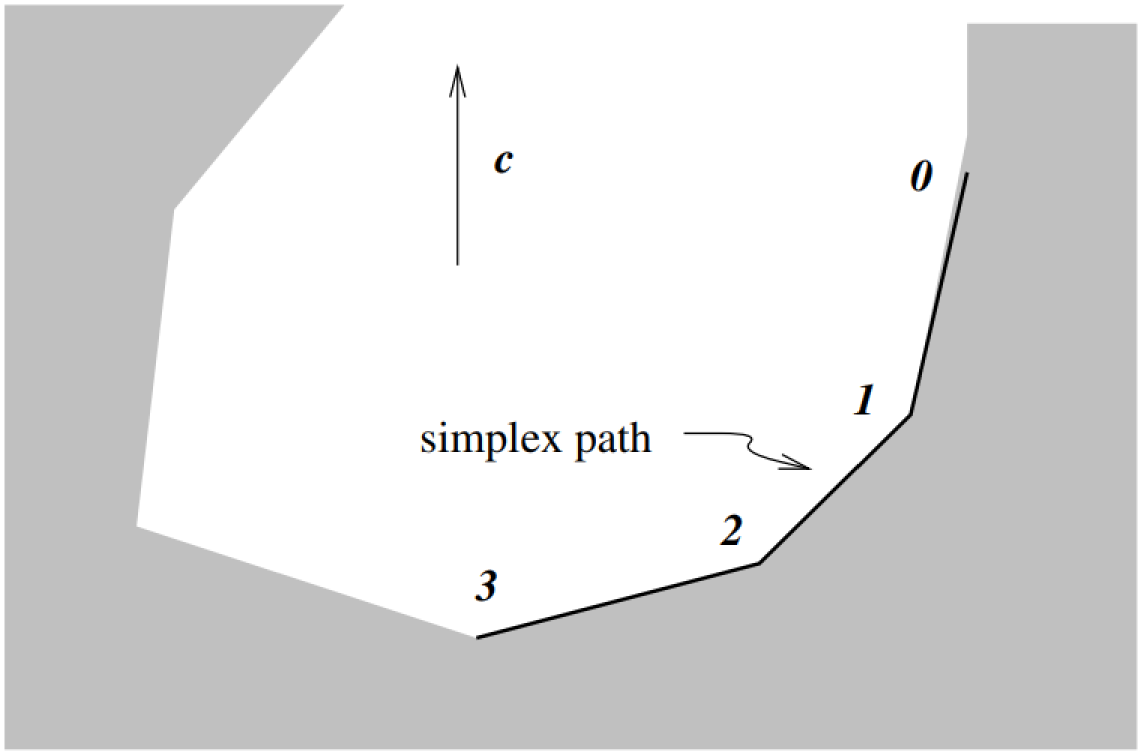
$\hookrightarrow p$!

$$\mathcal{N} \underset{p}{\overset{q}{\rightleftarrows}} \mathcal{B}$$

Repeat



if $q = 1$



c
0
1
simplex path
2
3

Check KKT-conditions for BFP

- Given BFP x, and corresponding basis $\mathcal{B}(x)$. Define

$$\mathcal{N}(x) = \{1, 2, \ldots, n\} \backslash \mathcal{B}(x)$$

- Partition x, s and c :

$$x_B = [x_i]_{i \in \mathcal{B}(x)} \qquad x_N = [x_i]_{i \in \mathcal{N}(x)}$$

- KKT conditions

KKT-2: $Ax = Bx_B + Nx_N = Bx_B = b$ (since *x* is BFP)

KKT-3: $x_B = B^{-1}b \geq 0, \quad x_N = 0$ (since *x* is BFP)

KKT-5: $x^\top s = x_B^\top s_B + x_N^\top s_N = 0$ if we choose $s_B = 0$

KKT-1: $\begin{bmatrix} B^T \\ N^T \end{bmatrix} \lambda + \begin{bmatrix} s_B \\ s_N \end{bmatrix} = \begin{bmatrix} c_B \\ c_N \end{bmatrix} \Rightarrow \begin{cases} \lambda = B^{-T} c_B \\ s_N = c_N - N^T \lambda \end{cases}$

KKT-4: Is $s_N \geq 0$?

- If $s_N \geq 0$, then the BFP x fulfills KKT and is a solution
- If not, change basis, and try again
  - E.g. pick smallest element of s_N (index q), increase x_q along $Ax=b$ until x_p becomes zero.
    Move q from $\mathcal{N}$ to $\mathcal{B}$, and p from $\mathcal{B}$ to $\mathcal{N}$.
    This guarantees decrease of objective, and no "cycling" (if non-degenerate).

Ex. 13.1

$$\min_{x \in \mathbb{R}^2} \; -4x_1 - 2x_2$$

s.t. $x_1 + x_2 \leq 5$

$2x_1 + \frac{1}{2}x_2 \leq 8$

$x_1 \geq 0, \; x_2 \geq 0$

standard form:

$$\min_{x \in \mathbb{R}^4} \; -4x_1 - 2x_2$$

s.t. $x_1 + x_2 + x_3 = 5$

$2x_1 + \frac{1}{2}x_2 + x_4 = 8$

$x_1 \geq 0, \; x_2 \geq 0, \; x_3 \geq 0, \; x_4 \geq 0$

$$c = \begin{bmatrix} -4 \\ -2 \\ 0 \\ 0 \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 2 & \frac{1}{2} & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 5 \\ 8 \end{bmatrix}$$

initial point $x = \begin{pmatrix} 0 \\ 0 \\ 5 \\ 8 \end{pmatrix}$, $\leftarrow$ feasible

$\mathcal{B}(x) = \{3, 4\}$

Ex. 13.1, first iteration

$$c^\top = (-4 \quad -2 \quad 0 \quad 0)\,, \quad A = \begin{pmatrix} \underbrace{\begin{matrix}1 & 1\\ 2 & .5\end{matrix}}_{N} & \underbrace{\begin{matrix}1 & 0\\ 0 & 1\end{matrix}}_{B} \end{pmatrix}, \quad b = \begin{pmatrix}5\\8\end{pmatrix}$$

$\mathcal{B} = \{3, 4\}$, $\mathcal{N} = \{1, 2\}$

$B = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$, $N = \begin{bmatrix}1 & 1\\ 2 & 0.5\end{bmatrix}$

$x_B = B^{-1} b = \begin{bmatrix}5\\8\end{bmatrix}$

$\lambda = (B^T)^{-1} c_B = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix} = \begin{bmatrix}0\\0\end{bmatrix}$

$\begin{bmatrix}s_1\\s_2\end{bmatrix} = s_N = c_N - N^T\lambda = \begin{bmatrix}-4\\-2\end{bmatrix} - \begin{bmatrix}1 & 2\\ 1 & 0.5\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix} = \begin{bmatrix}-4\\-2\end{bmatrix}$ $(\neq 0\,!)$

- pick smallest element of s_N: $s_1 = -4 \to q = 1$
- Let $q=1$ enter $\mathcal{B}$ (leaving $\mathcal{N}$)
- Increase x_1 while $Ax=b$. x_4 becomes zero first.
- Let $p=4$ leave $\mathcal{B}$ and enter $\mathcal{N}$.

Ex. 13.1, second iteration

$c^\top = (-4 \quad -2 \quad 0 \quad 0)\,,\ A = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 2 & .5 & 0 & 1 \end{pmatrix},\ b = \begin{pmatrix} 5 \\ 8 \end{pmatrix}$

$\mathcal{B} = \{1, 3\}\,,\ \mathcal{N} = \{2, 4\}$

$B = \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix},\quad B^{-1} = \frac{1}{2}\begin{bmatrix} 0 & 1 \\ 2 & -4 \end{bmatrix},\quad N = \begin{bmatrix} 1 & 0 \\ 1/2 & 1 \end{bmatrix}$

$x_N = \begin{bmatrix} 0 \\ 0 \end{bmatrix},\quad x_B = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} = B^{-1} b = \begin{bmatrix} 4 \\ 1 \end{bmatrix}$

$\lambda = (B^T)^{-1} c_B = \frac{1}{2}\begin{bmatrix} 0 & 2 \\ 1 & -4 \end{bmatrix}\begin{bmatrix} -4 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -2 \end{bmatrix}$

$s_N = c_N - N^T \lambda = \begin{bmatrix} -2 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 & 1/2 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ -2 \end{bmatrix} = \begin{bmatrix} -1 \\ 2 \end{bmatrix} = \begin{bmatrix} s_2 \\ s_4 \end{bmatrix}$

$s_2 = -1 \leq 0 \rightarrow \underline{q = 2}$

$q = 2$: entering index

$p = 3$: leaving index

Ex. 13.1, third iteration

$$c^\top = \begin{pmatrix} -4 & -2 & 0 & 0 \end{pmatrix}, \quad A = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 2 & .5 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 5 \\ 8 \end{pmatrix}$$

$$\mathcal{B} = \{1, 2\}, \quad \mathcal{N} = \{3, 4\} \qquad B = \begin{bmatrix} 1 & 1 \\ 2 & \frac{1}{2} \end{bmatrix} \qquad N = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$B^{-1} = \frac{1}{3} \begin{bmatrix} -1 & 2 \\ 4 & -2 \end{bmatrix}$$

$$x_N = \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

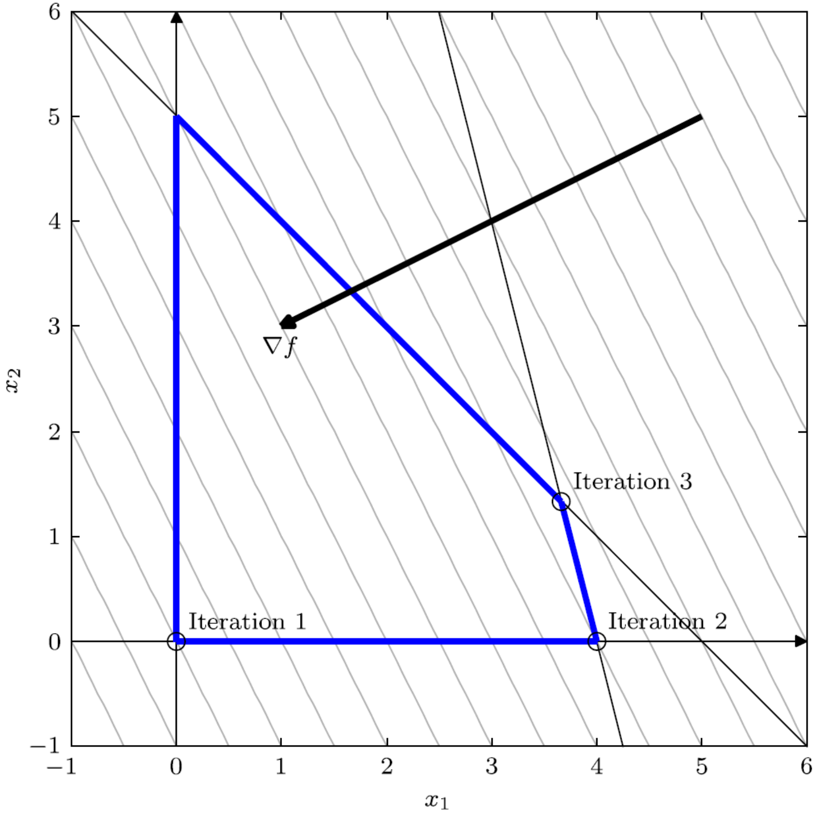
$$x_B = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = B^{-1} b = \begin{bmatrix} 11/3 \\ 4/3 \end{bmatrix}$$

$$\lambda = (B^{-1})^T c_B = \ldots = \begin{bmatrix} -4/3 \\ -4/3 \end{bmatrix}$$

$$s_N = c_N - N^T \lambda = \begin{bmatrix} 4/3 \\ 4/3 \end{bmatrix} \qquad \geq 0 \, !$$

Solution: $x^* = \begin{bmatrix} 11/3 \\ 4/3 \\ 0 \\ 0 \end{bmatrix}$!

Example 13.1 – figure



Linear algebra – LU factorization

- Two linear systems must be solved in each iteration:
  - $B^\top \lambda = c_B$
  - $Bd = A_q$ (to find the direction to check when increasing x_q)
  - We also have $Bx_B = b$. Since x_B is not needed in the iterations, we don't need to solve this (apart from in the final iteration)
  - This is the major work per iteration of simplex, efficiency is important!

- *B* is a general, non-singular matrix
  - Guaranteed a solution to the linear systems
  - LU factorization is the appropriate method to use (same for both systems)
  - Don't use matrix inversion!

- In each step of Simplex method, one column of *B* is replaced:
  - Can update ("maintain") the LU factorization of *B* in a smart and efficient fashion
  - No need to do a new LU factorization in each step, save time!

Starting the Simplex method

- We assumed an initial BFP available – but finding this is as difficult as solving the LP
- Normally, simplex algorithms have two phases:
  - Phase I: Find BFP
  - Phase II: Solve LP

- Phase I: Design other LP with trivial initial BFP, and whose solution is BFP for original problem

$$\min e^\top z \text{ subject to } Ax + Ez = b, \quad (x, z) \geq 0$$

$$e = (1, 1, \ldots, 1)^\top, \quad E \text{ diagonal matrix with } \begin{cases} E_{jj} = 1 \text{ if } b_j \geq 0 \\ E_{jj} = -1 \text{ if } b_j < 0 \end{cases}$$

Other practical implementation issues (Ch. 13.5)

- Selection of “entering index” q
  - Dantzig’s rule: Select the index of the most negative element in s_N
  - Other rules have proved to be more efficient in practice
- Handling of degenerate bases/degenerate steps (when increasing x_q is not possible)
  - If no degeneracy, each step leads to decrease in objective and convergence in finite number of iterations is guaranteed (Thm 13.4)
  - Degenerate steps lead to no decrease in objective. Not necessarily a problem, but can lead to cycling (we end up in the same basis as before)
  - Practical algorithms use perturbation strategies to avoid this
- Presolving (Ch. 13.7)
  - Reducing the size of the problem before solving, by various tricks to eliminate variables and constraints. Size reduction can be huge. Can also detect infeasibility.

Simplex complexity

- Typically, at most *2m* to *3m* iterations
- Worst case: All vertices must be visited (exponential complexity in *n*)
- Compare interior point method: Guaranteed polynomial complexity, but in practice hard to beat simplex on many problems

Simplex – an active set method

- Active set methods (such as simplex method):
  - Are iterative algorithms: Needs a starting point, and iterates several steps before it (hopefully) ends up in a solution: a point that fulfills the KKT conditions
  - Maintains explicitly an estimate of the set of inequality constraints that are active at the solution (the set $\mathcal{N}$ for the simplex method)
  - Makes small changes to the set in each iteration (a single index in simplex)

- Next time: Active set method for QP