Network Algorithms

Chapter 3

Optimization Methods

- Problem Definition & Notation
- Structure of the Feasible Set
- The Simplex Algorithm
- Simplex Tableaus
- Duality
- Branch & Bound
- Cutting Planes

(Dieser Foliensatz wurde von Herrn Prof. Dr. Thomas Böhme inhaltlich ausgearbeitet.)

Optimization Problems

- An *optimization problem* is given by a set M and a function $f : X \to \mathbb{R}$ with $X \supseteq M$.
- The usual terminology is as follows:
  - M is called the *feasible set*.
  - f is called the *objective function*
  - Elements of M are called *feasible solutions*.
  - $x^* \in M$ is an *optimal (maximal) solution* if $f(x^*) \geq f(x)$ for all $x \in M$, i.e.
  $$f(x^*) = \max\{f(x) \mid x \in M\}.$$
- An optimization method is an algorithm that computes an optimal solution x^* given the input (M, f) if there is any.
- Note that $\min\{f(x) \mid x \in M\} = -\max\{-f(x) \mid x \in M\}$. Thus, there is no need to deal with minimization problems separately.

Linear Optimization Problems (LOP)

- An optimization problem (M, f) is a *linear optimization problem (LOP)* if $M \subseteq \mathbb{R}^n$ for some $n \in \mathbb{N}$ consists of all $\vec{x} \in \mathbb{R}^n$ satisfying a finite set of linear inequalities and/or linear equations, and $f : \mathbb{R}^n \to \mathbb{R}$ is a linear function.

- Example 1:

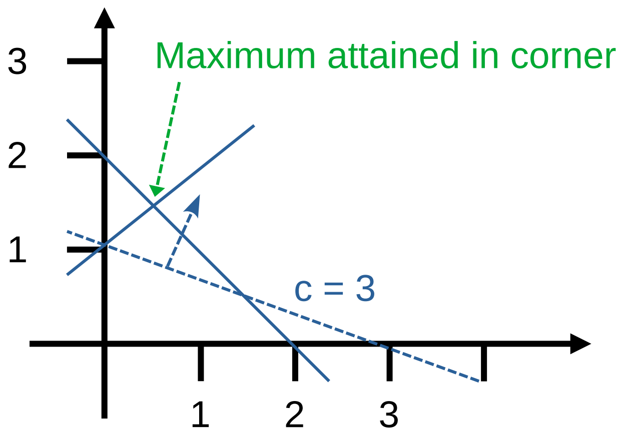
  Maximize

$$f(x, y) = x + 3y$$

  subject to

$$\begin{aligned} -x + y &\leq 1 \\ x + y &\leq 2 \\ x, y &\geq 0 \end{aligned}$$

Solving LOPs With Two Variables Graphically



$x + y \leq 2 \Leftrightarrow y \leq 2 - x$

$-x + y \leq 1 \Leftrightarrow y \leq 1 + x$

$x, y \geq 0$

Max. $f(x, y) = x + 3y$

- Finding the optimal solution by parallel shifting of the contour lines:
  - The equality c = f(x, y) defines for arbitrary c the so-called contour lines (German: “Höhenlinien”) of the objective function.
  - By increasing the value of c – which corresponds to a parallel shift of this line – up to the point where the line would leave the feasible set, an optimal solution can be found. One such solution is always found in a corner of the feasible set.
- Note:
  - It is not obvious in which corner the optimal solution can be found.
  - Depending on the constraints there may be many corners.

Some Remarks on Notation

- $\mathbb{R}^n$ is the set of all column vectors $\vec{x} = (x_1, \ldots, x_n)^T$ with $x_1, \ldots x_n \in \mathbb{R}$.
- We write $\vec{a} \leq \vec{b}$ if and only if $a_k \leq b_k$ for all $k \in \{1, \ldots, n\}$, and use this notation for row vectors correspondingly.
  (Caution: This is not a linear order. For example, neither $(1,2)^T \leq (2,1)^T$ nor $(1,2)^T \geq (2,1)^T$ holds.)
- $\vec{o}$ denotes a zero vector (with appropriately many entries).
- I denotes an identity matrix (with appropriately many rows and columns).
- $||\vec{x}|| = \sqrt{x_1^2 + \cdots + x_n^2}$ denotes the euclidean norm of $\vec{x}$.

Canonical Form LOPs

Canonical Form:

Maximize

$$f(\vec{x}) = \vec{c}^{\,T}\vec{x}$$

subject to

$$A\vec{x} \leq \vec{b}, \quad \vec{x} \geq \vec{o}$$

Here, $A \in \mathbb{R}^{m \times n}$ is real matrix with m rows and n columns and $\vec{b} \in \mathbb{R}^m$ is a real column vector with m entries.

Canonical Form LOPs

- If there is a variable x_k subject to $x_k \leq 0$ replace every appearance of x_k with $-x_k$ and replace $x_k \leq 0$ with $x_k \geq 0$.
- If a variable x_k is unrestricted, replace every appearance of x_k by $x_k^+ - x_k^-$ and let $x_k^+ \geq 0$, $x_k^- \geq 0$.
  This results in an equivalent LOP where each variable is non-negative.
- In the following, let $\underline{a}_k$ be the k_{th} row vector of the matrix A.
  The inequality $\underline{a}_k\ \vec{x} \leq b_k$ is equivalent to $-\underline{a}_k\ \vec{x} \geq -b_k$, and the equality $\underline{a}_k\ \vec{x} = b_k$ is equivalent to ($\underline{a}_k\ \vec{x} \leq b_k$ and $\underline{a}_k\ \vec{x} \geq b_k$).
  Thus, any LOP can be transformed into an equivalent LOP in canonical form.

Standard Form LOPs

- **Standard Form:**

  Maximize

$$f(\vec{x}) = \vec{c}^T \vec{x}$$

  subject to

$$A\vec{x} = \vec{b}$$

  and

$$\vec{x} \geq \vec{o}$$

  Here, $A \in \mathbb{R}^{m \times n}$ is a matrix with rank m and $\vec{b} \in \mathbb{R}^m$ is a column vector with $\vec{b} \geq \vec{o}$.

Transformation into Standard Form (1)

- Assume the LOP is given in canonical form, i.e.the feasible set M is written as $M = \{\vec{x} \in \mathbb{R}^n \mid (A\vec{x} \le \vec{b}) \wedge (\vec{x} \ge \vec{o})\}$ such that $A \in \mathbb{R}^{m \times n}$, $\vec{b} \in \mathbb{R}^m$.
- Clearly,

$$M = \{\vec{x} \in \mathbb{R}^n \mid \exists \vec{y} \in \mathbb{R}^m : \ (A\vec{x} + \vec{y} = \vec{b}) \wedge (\vec{x} \ge \vec{o}) \wedge (\vec{y} \ge \vec{o}))\}.$$

  The additional variables in $\vec{y}$ are called *slack variables*.
- Note that the equality $A\vec{x} + \vec{y} = \vec{b}$ can be re-written as

$$(A|I)\begin{pmatrix} \vec{x} \\ \vec{y} \end{pmatrix} = \vec{b}.$$

- If $\vec{b}$ does not satisfy $\vec{b} \ge \vec{o}$, multiply all rows k with $b_k < 0$ by (-1).

Transformation into Standard Form (2)

- Finally, recall that if $A \in \mathbb{R}^{m \times n}$, then $rk(A) \le m$, and that if $rk(A) < m$, then the system of linear equations $A\vec{x} = \vec{b}$ either has no solution at all or there are $m - rk(A)$ redundant equations. Consequently, the assumption that $rk(A) = m$ is no restriction.
- Example for linear dependent (and thus redundant) rows:

$$\begin{pmatrix} 1 & 2 & -1 \\ 3 & 1 & 0 \\ 4 & 3 & -1 \\ 8 & 6 & -2 \end{pmatrix} \vec{x} = \begin{pmatrix} 3 \\ 4 \\ 7 \\ 14 \end{pmatrix}$$

The Structure of the Feasible Set (1)

Definition: A subset X of $\mathbb{R}^n$ is called

- *convex* if for any two points $\vec{x}_1, \vec{x}_2 \in X$ the whole straight line segment $\overline{\vec{x}_1\vec{x}_2} = \{\alpha\vec{x}_1 + (1-\alpha)\vec{x}_2 \mid \alpha \in [0,1]\}$ is in X,
- *closed* if the limit of every convergent sequence in X is in X too, and
- *bounded* if there is $K \in \mathbb{R}$ such that $\|\vec{x}\| \le K$ for all $\vec{x} \in X$.

The Structure of the Feasible Set (2)

We recall the following statements.

- The intersection of two convex sets is convex.
- The intersection of two closed sets is closed.
- If $f : X \to \mathbb{R}$ is continuous and $M \subseteq X$ is nonempty, closed and bounded, then there is a $x^* \in M$ such that $\forall x \in M : f(x) \le f(x^*)$ i.e. f attains its maximum on M.
- If $\vec{d} \in \mathbb{R}^n$ and $t \in \mathbb{R}$, then the set $\{\vec{x} \in \mathbb{R}^n \mid \vec{d}^T\vec{x} \le t\}$ is convex and closed.
- Linear functions from $\mathbb{R}^n$ into $\mathbb{R}$ are continuous.

The Structure of the Feasible Set (3)

- Consider a LOP in standard form:
  Maximize
  $$f(\vec{x}) = \vec{c}^T \vec{x}$$
  subject to
  $$A\vec{x} = \vec{b}$$
  $$\vec{x} \geq \vec{o}$$
  with $A \in \mathbb{R}^{m \times n}$, $\vec{b} \geq \vec{o}$ and $rk(A) = m$.
- By the statements from the previous slide we deduce that the feasible set $M = \{\vec{x} \in \mathbb{R}^n \mid (A\vec{x} = \vec{b}) \wedge (\vec{x} \geq \vec{o})\}$ is convex and closed.
- If M is also nonempty and bounded, f attains its maximum on M.

Extreme Points and Basic Solutions (1)

Definition: Let M be a convex set. A point $\vec{x}_0 \in M$ is called an *extreme point* of M if $\vec{x}_0$ cannot be expressed as a convex linear combination $\alpha\vec{x}_1 + (1-\alpha)\vec{x}_2$ with $\alpha \in [0,1]$ of two points $\vec{x}_1, \vec{x}_2 \in M$ with $\vec{x}_1 \neq \vec{x}_0 \neq \vec{x}_2$.

We mention the following statements without proof.

(A) If the set $M = \{\vec{x} \in \mathbb{R}^n \mid (A\vec{x} = \vec{b}) \wedge (\vec{x} \geq \vec{o})\}$ is nonempty, then M has at most finitely many and at least one extreme point.

(B) If M is in addition bounded, then M is the convex hull of its extreme points $\vec{x}_1, \ldots, \vec{x}_k$,
i.e. any point $\vec{x}$ in M can be expressed as convex linear combination $\vec{x} = \sum_{i=1}^{k} \alpha_k \vec{x}_i$ with $\alpha_1, \ldots, \alpha_k \in [0,1]$ and $\sum_{i=1}^{k} \alpha_i = 1$.

Extreme Points and Basic Solutions (2)

Theorem 1 *If M is nonempty and bounded, then there is an extreme point $\vec{x}^* \in M$ such that $f(\vec{x}^*) \geq f(\vec{x})$ for all $\vec{x} \in M$.*

Proof:

Since f is linear and M is closed and bounded, there is a point $\vec{x}_0 \in M$ such that $f(\vec{x}_0) \geq f(\vec{x})$ for all $\vec{x} \in M$.

Since M is nonempty and bounded, $\vec{x}_0$ can be expressed as convex linear combination $\vec{x}_0 = \sum_{i=1}^{k} \alpha_i \vec{x}_i$ of extreme points $\vec{x}_1, \ldots, \vec{x}_k$ with $\alpha_1, \ldots, \alpha_k \in [0,1]$ and $\sum_{i=1}^{k} \alpha_i = 1$.

By linearity of f it is $f(\vec{x}_0) = \sum_{i=1}^{k} \alpha_i f(\vec{x}_i) \leq \max\{f(\vec{x}_1), \ldots, f(\vec{x}_k)\}$.

By maximality of $f(\vec{x}_0)$ it is $f(\vec{x}_0) \geq \max\{f(\vec{x}_1), \ldots, f(\vec{x}_k)\}$.

Hence, there is an $i \in \{1, \ldots, k\}$ such that $f(\vec{x}_0) = f(\vec{x}_i)$.

□

Extreme Points and Basic Solutions (3)

If M is nonempty and unbounded there need not be an optimal solution.

However, if there is one, a similar but more elaborate argument (not given here) can be used to prove the following theorem.

Theorem 2 *If M is nonempty and there exists an K such that $f(\vec{x}) \leq K$ for all $\vec{x} \in M$, then there is an extreme point $\vec{x}^* \in M$ such that $f(\vec{x}^*) \geq f(\vec{x})$ for all $\vec{x} \in M$.*

Extreme Points and Basic Solutions (4)

Definition:

- Let $A \in \mathbb{R}^{m \times n}$ be a matrix with column vectors $\vec{a}_1, \ldots, \vec{a}_n$,
- $M = \{\vec{x} \in \mathbb{R}^n \mid (A\vec{x} = \vec{b}) \wedge (\vec{x} \geq \vec{o})\}$, and
- $S(\vec{x}) = \{\vec{a}_i \mid (i \in \{1, \ldots, n\}) \wedge (x_i \neq 0)\}$ for $\vec{x} \in M$.

Then $\vec{x}$ is called a *basic solution* if $S(\vec{x})$ is linear independent.

Note that linear independence of $S(\vec{x})$ is equivalent to the assertion that there is no $\vec{v} \neq \vec{o}$ with the property that $v_i \neq 0$ implies $x_i \neq 0$ such that $A\vec{v} = \vec{o}$.

Theorem 3 *Let $\vec{x} \in M$. Then the following two statements are equivalent.*

(a) $\vec{x}$ is an extreme point of M.

(b) $\vec{x}$ is a basic solution.

Extreme Points and Basic Solutions (5)

Proof of Theorem (3). [Contraposition: $(P \implies Q) \iff (\neg Q \implies \neg P)$]

(a) $\implies$ (b): Suppose that $\vec{x}$ is not a basic solution,

i.e. $S(\vec{x})$ is not linear independent. Then there is a vector $\vec{v} \neq \vec{o}$ with $v_i \neq 0$ implies $x_i \neq 0$ for all $i \in \{1, \ldots, n\}$ such that $A\vec{v} = \vec{o}$.

Thus, there is an $\varepsilon > 0$ such that $\vec{u}_1 = \vec{x} + \varepsilon\vec{v} \geq \vec{o}$ and $\vec{u}_2 = \vec{x} - \varepsilon\vec{v} \geq \vec{o}$.

Clearly, $\vec{u}_1, \vec{u}_2 \in M$, $\vec{u}_1 \neq \vec{x} \neq \vec{u}_2$, and $\frac{1}{2}\vec{u}_1 + \frac{1}{2}u_2 = \vec{x}$, i.e. $\vec{x}$ is not an extreme point.

(b) $\implies$ (a): Suppose that $\vec{x}$ is not an extreme point.

Then there are vectors $\vec{v}_1, \vec{v}_2 \in M$ such that $\vec{v}_1 \neq \vec{x} \neq \vec{v}_2$ and an $\alpha \in (0,1)$ such that $\vec{x} = \alpha\vec{v}_1 + (1-\alpha)\vec{v}_2$.

It follows that $\vec{v}_1 \neq \vec{v}_2$, $A(\vec{v}_1 - \vec{v}_2) = \vec{o}$, and that if the i-th entry of $\vec{v}_1 - \vec{v}_2$ is not 0, then $x_i \neq 0$ (*, see also next slide).

This implies that $S(\vec{x})$ is not linear independent, and so $\vec{x}$ is not a basic solution. □

Extreme Points and Basic Solutions (6)

(*) Why does the i-th entry of $\vec{v}_1 - \vec{v}_2$ is not 0, imply that $x_i \neq 0$?

Suppose that $x_i = 0$:

As $\vec{x} = \alpha\vec{v}_1 + (1-\alpha)\vec{v}_2$ we obtain:

$$x_i = 0 = \alpha(v_{1,i} - v_{2,i}) + v_{2,i}$$

$$\iff \alpha(v_{2,i} - v_{1,i}) = v_{2,i}$$

$$\iff \alpha = \frac{v_{2,i}}{(v_{2,i} - v_{1,i})}$$

We have both $v_{1,i} \geq 0$ and $v_{2,i} \geq 0$. Thus, if $v_{2,i} \geq v_{1,i}$ then $\alpha > 1$. Furthermore, if $v_{2,i} \leq v_{1,i}$ then $\alpha < 0$. But, we required $\alpha \in (0,1)$.

Therefore, $x_i = 0$ clearly leads to a contradiction, so that $x_i \neq 0$ whenever the i-th entry of $\vec{v}_1 - \vec{v}_2$ is not 0. □

The Simplex Algorithm – Overview (1)

We consider the LOP in standard form (L) as follows.

Maximize

$$f(\vec{x}) = \vec{c}^T\vec{x}$$

subject to

$$A\vec{x} = \vec{b}$$

and

$$\vec{x} \geq \vec{o}$$

with

$$A \in \mathbb{R}^{m\times n}, \vec{b} \geq \vec{o}, \text{ and } rk(A) = m.$$

Let $M = \{\vec{x} \in \mathbb{R}^n \mid (A\vec{x} = \vec{b}) \wedge (\vec{x} \geq \vec{o})\}$.

The Simplex Algorithm – Overview (2)

We know so far:

- M has at least one extreme point.
- Any extreme point of M is a basic solution of (L) and vice versa.
- (L) has at most $\binom{n}{m}$ basic solutions.
- If there is an optimal solution of (L), then there is an optimal solution of (L) that is a basic solution.
- If M is nonempty and bounded, then there is an optimal solution of (L).

It follows that (at least) if M is nonempty and bounded, there is a finite algorithm that finds an optimal solution of (L).

The Simplex Algorithm – Overview (3)

- The simplex algorithm consists of two parts, called Phase 1 and Phase 2.
- The input of Phase 2 is a feasible basic solution.
- The simplex algorithm stops when either an optimal basic has been found, or if it has been detected that the objective function is unbounded on M. In the latter case there is no optimal solution of (L).
- Phase 1 is needed only if there is no feasible basic solution known.
- Phase 1 consists in applying Phase 2 to an auxiliary LOP. Phase 1 stops when a feasible basic solution of (L) has been found or if it has been detected that M is empty.

The Simplex Algorithm – Phase 2 (1)

- Let $\vec{x}$ be a feasible basic solution of (L).
- Let $T(\vec{x}) \subseteq \{\vec{a}_1, \ldots, \vec{a}_n\}$ be a maximal linear independent set of column vectors of A such that

$$x_k \neq 0 \implies \vec{a}_k \in T(\vec{x}).$$

(Note: Since $rk(A) = m$ the set $T(\vec{x})$ has m elements.)

- Let $B = \{k \in \{1, \ldots, n\} \mid \vec{a}_k \in T(\vec{x})\}$ and $N = \{1, \ldots, n\} \setminus B$.
- If $T(\vec{x})$ contains column vectors $\vec{a}_k$ such that $x_k = 0$, the basic solution $\vec{x}$ is called *degenerate*, otherwise it is said to be *non-degenerate*.

The Simplex Algorithm – Phase 2 (2)

- $T(\vec{x})$ is a basis of the linear subspace generated by all column vectors of A.
  Thus, each column vector $\vec{a}_j$ can represented as a linear combination of the elements of $T(\vec{x})$, and this representation is unique:

$$\vec{a}_j = \sum_{k \in B} t_{k,j} \vec{a}_k.$$

- Clearly, if $j \in B$, then $t_{j,j} = 1$ and $t_{k,j} = 0$ if $k \neq j$.
- For $j \in N$ let $u_j = \sum_{k \in B} t_{k,j} c_k$ and $d_j = u_j - c_j$.
  - Recall that the objective function f is given by $f(\vec{x}) = \vec{c}^T \vec{x}$ with $\vec{c} = (c_1, \ldots, c_n)^T$.

The Simplex Algorithm – Phase 2 (3)

Let $\vec{y} \in \mathbb{R}^n$ be an arbitrary feasible solution of (L), i.e. $\sum_{i=1}^{n} y_i \vec{a}_i = \vec{b}$ and $\vec{y} \geq \vec{o}$. Then

$$\begin{aligned} \vec{b} &= \sum_{k \in B} x_k \vec{a}_k = \sum_{i=1}^{n} y_i \vec{a}_i = \sum_{i=1}^{n} y_i (\sum_{k \in B} t_{k,i} \vec{a}_k) \\ &= \sum_{k \in B} (\sum_{i=1}^{n} t_{k,i} y_i) \vec{a}_k. \end{aligned}$$

Since $T(\vec{x})$ is linear independent we know that for all $k \in B$

$$x_k = \sum_{i=1}^{n} t_{k,i} y_i = y_k + \sum_{i \in N} t_{k,i} y_i.$$

The Simplex Algorithm – Phase 2 (4)

Consequently, $y_k = x_k - \sum_{i \in N} t_{k,i} y_i$ for all $k \in B$.

We can now express $f(\vec{y}) = \vec{c}^T \vec{y}$ as follows

$$\begin{aligned} \vec{c}^T \vec{y} &= \sum_{i=1}^{n} c_i y_i \\ &= \sum_{k \in B} c_k y_k + \sum_{i \in N} c_i y_i \\ &= \sum_{k \in B} c_k (x_k - \sum_{i \in N} t_{k,i} y_i) + \sum_{i \in N} c_i y_i \\ &= \sum_{k \in B} c_k x_k + \sum_{i \in N} (c_i - \sum_{k \in B} c_k t_{k,i}) y_i \\ &= \vec{c}^T \vec{x} - \sum_{i \in N} d_i y_i \end{aligned}$$

The Simplex Algorithm – Phase 2 (5)

We distinguish three cases:

<u>Case 1:</u> It is $d_i \geq 0$ for all $i \in N$.

Then $f(\vec{x}) \geq f(\vec{y})$ for all feasible solutions $\vec{y}$, i.e. $\vec{x}$ is an optimal solution.

In this case the algorithm stops.

<u>Case 2:</u> There is an index $j \in N$ such that $d_j < 0$ and $t_{k,j} \leq 0$ for all $k \in B$.

Let $\alpha > 0$. Then the vector $\vec{z} \in \mathbb{R}^n$ with

$$z_k = \begin{cases} x_k - \alpha t_{k,j} & \text{for } k \in B \\ \alpha & \text{for } k = j \\ 0 & \text{for } k \in N \text{ and } k \neq j \end{cases}$$

is nonnegative, and: (see next slide)

The Simplex Algorithm – Phase 2 (6)

... is nonnegative, and:

$$A\vec{z} = \sum_{k=1}^{n} z_k \vec{a}_k = \sum_{k \in B} x_k \vec{a}_k - \alpha \sum_{k \in B} t_{k,j} \vec{a}_k + \alpha \vec{a}_j =$$

$$= \sum_{k \in B} x_k \vec{a}_k - \alpha \vec{a}_j + \alpha \vec{a}_j = \sum_{k \in B} x_k \vec{a}_k = A\vec{x} = \vec{b}.$$

Consequently, $\vec{z}$ is a feasible solution for all $\alpha > 0$. Since $d_j < 0$,

$$\lim_{\alpha \to \infty} f(\vec{z}) = \lim_{\alpha \to \infty} f(\vec{x}) - d_j \alpha = \infty,$$

i.e. the objective function f is unbounded from above on M.

In this case the algorithm stops.

The Simplex Algorithm – Phase 2 (7)

Case 3 There is an index $j \in N$ such that $d_j < 0$ and there is an index $\ell \in B$ such that $t_{\ell,j} > 0$.

Let $\varepsilon = \min\{x_k/t_{k,j} \mid (k \in B) \wedge (t_{k,j} > 0)\}$. Clearly, $\varepsilon \geq 0$.

Suppose that $\varepsilon = x_\ell/t_{\ell,j}$ and define $\vec{z} \in \mathbb{R}^n$ by

$$z_k = \begin{cases} x_k - \varepsilon t_{k,j} & \text{for } k \in B \\ \varepsilon & \text{for } k = j \\ 0 & \text{for } k \in N \text{ and } k \neq j \end{cases} .$$

Similarly as in Case 2 it is not hard to see that $\vec{z}$ is a feasible solution.

Furthermore, $z_\ell = 0$ and since $t_{\ell,j} \neq 0$, the j-th column vector $\vec{a}_j$ of A cannot be expressed as a linear combination of the column vectors in $S(\vec{x}) \setminus \{\vec{a}_\ell\}$.

Thus, $\vec{z}$ is a feasible basic solution.

The Simplex Algorithm – Phase 2 (8)

By $d_j < 0$ and $\varepsilon > 0$, it is $f(\vec{z}) = f(\vec{x}) - d_j\varepsilon \geq f(\vec{x})$.

The algorithm starts over with the new feasible basic solution $\vec{z}$.

Special care is needed to avoid ‘cycling’.
One way to avoid cycling is to apply Bland’s rule (in Case 3 above)[1]

Bland’sRule :

- Let $j = \min\{i \in N \mid d_i < 0\}$.
- Let $\ell = \min\{s \in B \mid x_s/t_{s,j} = \varepsilon\}$, with ε as above.

1: A proof that Bland’s rule indeed prevents cycling can be found in: J.Matousek and B. Gärtner, *Understanding and Using Linear Programming*, Springer, 2006.

The Simplex Algorithm – Phase 1 (1)

If the original LOP is given in standard from (L), and no feasible basic solution is known, an auxiliary LOP can be used to find one.

The Auxiliary LOP

Minimize $\quad g(\vec{x},\vec{y}) = \sum_{i=1}^{m} y_i$

subject to

$$A\vec{x} + \vec{y} = \vec{b}, \quad \vec{x} \geq \vec{o}, \quad \vec{y} \geq \vec{o}.$$

A feasible basic solution for the auxiliary problem is $\vec{y} = \vec{b}, \quad \vec{x} = \vec{o}$.

Thus, the feasible set of the auxiliary LOP is nonempty.

The Simplex Algorithm – Phase 1 (2)

Theorem 4

(a) The auxiliary LOP has an optimal solution $(\vec{y}^*, \vec{x}^*)$.

(b) The original LOP has a feasible solution if and only if $\vec{y}^* = \vec{o}$.

Proof:

(a) Since $\vec{y} \geq \vec{o}$, we have $g(\vec{x},\vec{y}) = \sum_{i=1}^{m} y_i \geq 0$.

Furthermore, the feasible set of the auxiliary LOP is nonempty.

It follows that the auxiliary LOP has an optimal solution.

(b) It is not hard to see that $\vec{x} = \vec{x}_0$ is a feasible solution of the original LOP if and only if $\vec{x} = \vec{x}_0, \vec{y} = \vec{o}$ is a feasible solution of the auxiliary one.

A feasible solution of the auxiliary LOP with $\vec{y} = \vec{o}$ is optimal because $g(\vec{x},\vec{o}) = 0$ for all $\vec{x} \in \mathbb{R}^n$. $\square$

Remark.

If the original LOP is given in canonical form with $\vec{b} \geq \vec{o}$,
then a feasible solution of the corresponding LOP in standard form

Maximize:

$$\vec{c}^T \vec{x}$$

subject to

$$A\vec{x} + I\vec{y} = \vec{b}$$

$$\vec{x} \geq \vec{o}$$

$$\vec{y} \geq \vec{o}$$

can be obtained as follows:

$$\vec{y} = \vec{b}, \quad \vec{x} = \vec{o}$$

Consider a system of linear equations $A\vec{x} = \vec{b}$.

Recall from linear algebra that $A\vec{x} = \vec{b}$ can be transformed into an equivalent system of linear equations $\hat{A}\vec{x} = \vec{\hat{b}}$ using the following ‘row operations’.

(1) Switching two equations.

(2) Multiplying one equation with a **nonzero** factor.

(3) Adding a multiple of an equation to **another** equation.

Here, ‘equivalent’ means that a vector $\vec{x}_0 \in \mathbb{R}^n$ is a solution of $A\vec{x} = \vec{b}$ if and only if it is a solution of $\hat{A}\vec{x} = \vec{\hat{b}}$.

Simplex Tableaus (2)

Now suppose, as above, that $A \in \mathbb{R}^{m \times n}$ with rank m, and $\vec{b} \in \mathbb{R}^m$.

Let $\vec{x}_0 \in \mathbb{R}^n$ be a feasible basic solution, i.e. $A\vec{x}_0 = \vec{b}$, $\vec{x}_0 \geq \vec{o}$, and the set $S(\vec{x}_0) = \{\vec{a}_i \mid x_{0i} \neq 0\}$ of column vectors of A is linear independent.

Let $T(\vec{x}_0) \supseteq S(\vec{x}_0)$ be a maximal linear independent set of column vectors containing $S(\vec{x}_0)$, $B = \{i \mid \vec{a}_i \in T(\vec{x}_0)\}$, and $N = \{1, \ldots, n\} \setminus B$.

Note that $|B| = m$ because $rank(A) = m$. In order to simplify our notation we assume henceforth that $B = \{1, \ldots, m\}$.

Using operations (1),(2), and (3), the system of linear equations $A\vec{x} = \vec{b}$ can be transformed into an equivalent system of linear equations $\hat{A}\vec{x} = \vec{\hat{b}}$ such that the first m columns of $\hat{A}$ form an $m \times m$ identity matrix I.

Thus $\hat{A}$ can be written as $(\hat{A}_B, \hat{A}_N) = (I, \hat{A}_N)$.

Note that this implies that $\vec{\hat{b}} = (\vec{x_0})_B$.

Simplex Tableaus (3)

Let $j \in N$ and express the column vector $\vec{a}_j$ as a linear combination of the column vectors in $T(\vec{x}_0)$

$$\vec{a}_j = \sum_{k \in B} t_{k,j} \vec{a}_k$$

Let $\vec{u} \in \mathbb{R}^n$ with

$$u_i = \begin{cases} t_{k,j} & \text{for } i = k \\ -1 & \text{for } i = j \\ 0 & \text{else} \end{cases}$$

Then $A\vec{u} = \sum_{i \in \{1,\ldots,n\}} u_i \vec{a}_i = \sum_{k \in B} t_{k,j} \vec{a}_k - \vec{a}_j = \vec{o}$.

It follows that $\hat{A}\vec{u} = \vec{o}$, as $\vec{o}$ does not change during the transformation.

Simplex Tableaus (3)

From the structure of $\hat{A}$ we see that

$$(\hat{A}_N)_{k,j} = t_{k,j}$$

for all $k \in B$ and $j \in N$, because

$$\hat{A} = (\hat{A}_B|\hat{A}_N) = (I_m|\hat{A}_N)$$

and

$$\begin{aligned}(I_m|\hat{A}_N)\begin{pmatrix}\vec{u}_B\\ \vec{u}_N\end{pmatrix} &= I_m\vec{u}_B + \hat{A}_N\vec{u}_N = \vec{o}\\ \iff \qquad \vec{u}_B &= -\hat{A}\vec{u}_N\\ k \in B &: (u_k = t_{k,j})\end{aligned}$$

Simplex Tableaus (4)

Since

$$t_{k,i} = \begin{cases} 1 & \text{for } k = i \\ 0 & \text{else} \end{cases}$$

for all $i, k \in B$, it finally follows that $(\hat{A})_{k,j} = t_{k,j}$ for all $k, j \in \{1, \ldots, n\}$.

Now we extend the system of linear equations $A\vec{x} = \vec{b}$ by appending the equation $\vec{c}^T\vec{x} = z$

$$\begin{pmatrix} A \\ \vec{c}^T \end{pmatrix}\vec{x} = \begin{pmatrix} \vec{b} \\ z \end{pmatrix}$$

This system of linear equation is equivalent to

$$\begin{pmatrix} \hat{A} \\ \vec{c}^T \end{pmatrix}\vec{x} = \begin{pmatrix} \vec{\hat{b}} \\ z \end{pmatrix}$$

Simplex Tableaus (5)

In this system of linear equations we add for each $k \in B$ the k-th equation multiplied by $-c_k$ to the last newly appended equation $\vec{c}^T\vec{x} = z$.

We get

$$\begin{pmatrix} \hat{A} \\ \vec{\hat{c}}^T \end{pmatrix} \vec{x} = \begin{pmatrix} \vec{\hat{b}} \\ \hat{z} \end{pmatrix}$$

where

$$\hat{c}_i = \begin{cases} 0 & \text{for } i \in B \\ c_i - \sum_{k \in B} c_k t_{k,i} & \text{for } i \in N \end{cases}$$

and

$$\hat{z} = z - \sum_{k \in B} c_k b_k$$

Note that $c_i - \sum_{k \in B} c_k t_{k,i} = -d_i$

(see also ‘The Simplex Algorithm – Phase 2 (2)’)

Simplex Tableaus (6)

Producing a new basic solution (Case 3 on ‘The Simplex Algorithm – Phase 2 (7)’) can now be described as follows.

Choose $j \in N$ such that $d_j < 0$.

Choose $\ell \in B$ such that $t_{\ell,j} > 0$ and $x_\ell / t_{\ell,j}$ is minimal.

If there are more than one candidate for j and ℓ, respectively, apply Bland’s rule.

Multiply the ℓ-th equation by $1/t_{\ell,j}$ and then, add the resulting ℓ-th equation multiplied by $-t_{k,j}$ to the k-th equation for $k \in \{1, \ldots, m\}$ with $k \neq \ell$.

Add the ℓ-th equation multiplied by $-\hat{c}_j = -d_j$ to the equation $\vec{c}^T\vec{x} = z$ giving $\vec{\hat{c}}^T\vec{x} = \hat{z}$.

Example 1

Example 1: (1)

$$\max \; x_1 + x_2$$

s.t.

$$\begin{aligned} -x_1 + x_2 &\le 1 \\ x_1 &\le 3 \\ x_2 &\le 2 \\ x_1, x_2 &\ge 0 \end{aligned}$$

Transform in Standard Form:

$$\max \; x_1 + x_2$$

s.t.

$$\begin{aligned} -x_1 + x_2 + y_1 &= 1 \\ x_1 + y_2 &= 3 \\ x_2 + y_3 &= 2 \\ x_1, x_2, y_1, y_2, y_3 &\ge 0 \end{aligned}$$

Initial Basic solution is easy to obtain:

$\begin{pmatrix} 0 \\ 0 \\ 1 \\ 3 \\ 2 \end{pmatrix}$ is feasible basic solution with $\vec{c}^T \cdot \vec{x} = 0$

So: (2)

$$\begin{array}{ccccc|cl} x_1 & x_2 & y_1 & y_2 & y_3 & b & \\ -1 & 1 & 1 & 0 & 0 & 1 & +1 \\ 1 & 0 & 0 & 1 & 0 & 3 & -1 \\ 0 & 1 & 0 & 0 & 1 & 2 & \\ \hline 1 & 1 & 0 & 0 & 0 & 0 & \\ \uparrow & & & & & & \end{array}$$

$$\begin{array}{ccccc|cl} 0 & 1 & 1 & 1 & 0 & 4 & -1 \\ 1 & 0 & 0 & 1 & 0 & 3 & \\ 0 & 1 & 0 & 0 & 1 & 2 & -1 \\ \hline 0 & 1 & 0 & -1 & 0 & -3 & \\ & \uparrow & & & & & \end{array}$$

$$\begin{array}{ccccc|c} 0 & 0 & 1 & 1 & -1 & 2 \\ 1 & 0 & 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & -1 & -1 & -5 \end{array}$$

Thus: $\begin{pmatrix} 3 \\ 2 \\ 2 \\ 0 \\ 0 \end{pmatrix}$ is optimal solution with $\vec{c}^T \cdot \vec{x} = 5$

Example 2

Example 2: (3)

$$\max \; x_1$$

s.t.

$$\begin{aligned} x_1 - x_2 &\le 1 \\ -x_1 + x_2 &\le 2 \\ x_1, x_2 &\ge 0 \end{aligned}$$

$$\begin{array}{cccc|cl} 1 & -1 & 1 & 0 & 1 & 1 \\ -1 & 1 & 0 & 1 & 2 & -1 \\ \hline 1 & 0 & 0 & 0 & 0 & \\ \uparrow & & & & & \end{array}$$

$$\begin{array}{cccc|c} 1 & -1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 3 \\ \hline 0 & 1 & -1 & 0 & -1 \\ & \uparrow & & & \end{array}$$

→ Case 2 ⇒ unbounded

Example 3

Example 3: (4)

$$\begin{array}{lrl} \max & x_2 & \\ \text{s.t.} & -x_1 + x_2 & \le 0 \\ & x_1 & \le 2 \\ & x_1, x_2 & \ge 0 \end{array}$$

Introduce slack variables:

$$\begin{array}{lrl} \max & x_2 & \\ \text{s.t.} & -x_1 + x_2 + x_3 & = 0 \\ & x_1 + x_4 & = 2 \\ & x_1, x_2, x_3, x_4 & \ge 0 \end{array}$$

$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 2 \end{pmatrix}$ is feasible basic solution with $\vec{c}^T \cdot x = 0$

(5)

$$\begin{array}{rrrr|rl} -1 & 1 & 1 & 0 & 0 & \\ 1 & 0 & 0 & 1 & 2 & -1 \\ \hline 0 & 1 & 0 & 0 & 0 & \end{array}$$

$$\begin{array}{rrrr|rl} -1 & 1 & 1 & 0 & 0 & +1 \\ 1 & 0 & 0 & 1 & 2 & -1 \\ \hline 1 & 0 & -1 & 0 & 0 & \\ \uparrow & & & & & \end{array}$$

$$\begin{array}{rrrr|r} 0 & 1 & 1 & 1 & 2 \\ 1 & 0 & 0 & 0 & 2 \\ \hline 0 & 0 & -1 & -1 & -2 \end{array}$$

Thus: $\begin{pmatrix} 2 \\ 2 \\ 0 \\ 0 \end{pmatrix}$ is optimal solution with $\vec{c}^T \cdot \vec{x} = 2$

Example 4 (1)

Example 4: (6)

$$\begin{array}{lrl} \max & x_1 + 2x_2 & \\ \text{s.t.} & x_1 + 3x_2 + x_3 & = 4 \\ & 2x_2 + x_3 & = 2 \\ & x_1, x_2, x_3 & \ge 0 \end{array}$$

Auxiliary Problem for Phase 1:

$$\begin{array}{lrl} \text{Max} & -y_1 - y_2 & \\ \text{s.t.} & x_1 + 3x_2 + x_3 + y_1 & = 4 \\ & 2x_2 + x_3 + y_2 & = 2 \\ & x_1, x_2, x_3, y_1, y_2 & \ge 0 \end{array}$$

$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 4 \\ 2 \end{pmatrix}$ is feasible basic solution with $\vec{c}^T \cdot \vec{x} = -6$

(7)

$$\begin{array}{rrrrr|rl} 1 & 3 & 1 & 1 & 0 & 4 & +1 \\ 0 & 2 & 1 & 0 & 1 & 2 & +1 \\ \hline 0 & 0 & 0 & -1 & -1 & 0 & \end{array}$$

$$\begin{array}{rrrrr|rl} 1 & 3 & 1 & 1 & 0 & 4 & \\ 0 & 2 & 1 & 0 & 1 & 2 & -1 \\ \hline 1 & 5 & 2 & 0 & 0 & 6 & \\ \uparrow & & & & & & \end{array}$$

$$\begin{array}{rrrrr|rl} 1 & 3 & 1 & 1 & 0 & 4 & -3/2 \\ 0 & 2 & 1 & 0 & 1 & 2 & -1 \quad 1:2 \\ \hline 0 & 2 & 1 & -1 & 0 & 2 & \\ & \uparrow & & & & & \end{array}$$

$$\begin{array}{rrrrr|r} 1 & 0 & -1/2 & 1 & -3/2 & 1 \\ 0 & 1 & 1/2 & 0 & 1/2 & 1 \\ \hline 0 & 0 & 0 & -1 & -1 & 0 \end{array}$$

$\begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$ is optimal solution with $\vec{c}^T \cdot x = 0$

Example 4 (2)

8

The tableau of the original LOP can be easily read from this:

$$
\begin{array}{ccc|c}
1 & 0 & -1/2 & 1 \\
0 & 1 & 1/2 & 1 \\
\hline
1 & 2 & 0 & 0
\end{array}
\quad \cdot(-1),\ \cdot(-2)
$$

$$
\begin{array}{ccc|c}
1 & 0 & -1/2 & 1 \\
0 & 1 & 1/2 & 1 \\
\hline
0 & 0 & -1/2 & -3
\end{array}
$$

Thus: $\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$ is optimal solution of original LOP with $\vec{c}^T \vec{x} = 3$

The Complexity of the Simplex Algorithm

- There are examples where the Simplex Algorithm starting with a specified feasible basic solution will pass through all vertices, and the number of vertices is exponential in the number n of variables.
- Yet, in practice the Simplex Algorithm is a very efficient approach. In many relevant cases, its running time is proportional to m+n.
- There are polynomial time algorithms for LOPs. (Khachian, Karmarkar)
- There are polynomial time variations of the Simplex Algorithm for special cases, e.g. single-commodity network flow problems.

Duality – Motivation (1)

①

Duality:

Consider the LOP (*)

$$\max f(\vec{x}) = 2x_1 + 3x_2$$

$$\begin{aligned} \text{s.t.} \quad & 4x_1 + 8x_2 \le 12 & \text{I.} \\ & 2x_1 + x_2 \le 3 & \text{II.} \\ & 3x_1 + 2x_2 \le 4 & \text{III.} \\ & x_1, x_2 \ge 0 \end{aligned}$$

As $x_1, x_2 \ge 0$, we have:

$$2x_1 + 3x_2 \le 4x_1 + 8x_2 \le 12$$

Thus, 12 is an upper bound for $f(\vec{x})$

If we divide the first equality by 2, we obtain an even better bound

$$2x_1 + 3x_2 \le 2x_1 + 4x_2 \le 6$$

Even better, if we calculate $1/3 \cdot (\text{I} + \text{II})$

$$2x_1 + 3x_2 \le 1/3(4x_1 + 8x + 2x_1 + x_2) \le 1/3(12+3) = 5$$

②

Hence, $f(\vec{x})$ can not get larger than 5

How good can an upper bound get this way?

So, we are trying to derive an inequality of the form

$$d_1 \cdot x_1 + d_2 \cdot x_2 \le h$$

with $d_1 \ge 2$, $d_2 \ge 3$ and h as small as possible

So, $\forall x_1, x_2 \ge 0$ we have

$$2x_1 + 3x_2 \le d_1 x_1 + d_2 x_2 \le h$$

Combine the 3 inequalities of the original LOP with some non-negative coefficients y_1, y_2, y_3 (so that direction of inequalities is not reversed), we obtain

$$\overbrace{(4y_1 + 2y_2 + 3y_3)}^{d_1} \cdot x_1 + \overbrace{(8y_2 + y_2 + 3y_3)}^{d_2} \cdot x_2$$

$$\le \underbrace{12y_1 + 3y_2 + 4y_3}_{h},$$

Duality – Motivation (2)

③

How to choose y_1, y_2, y_3?

We need to ensure that $d_1 \ge 2$ and $d_2 \ge 3$ and we want to have h as small as possible under these constraints

So, we want: (**)

$$\begin{aligned} \min \quad & 12y_1 + 3y_2 + 4y_3 \\ \text{s.t.} \quad & 4y_1 + 2y_2 + 3y_3 \ge 2 \\ & 8y_1 + y_2 + y_3 \ge 3 \\ & y_1, y_2, y_3 \ge 0 \end{aligned}$$

This is called the dual LOP to the original LOP (*)

④

The dual LOP guards the original LOP from above and the original LOP guards the dual LOP from below.

That is: Every feasible solution to (**) gives an upper bound to problem (*), and every feasible solution to (*) gives a lower bound to problem (**).

We obtain:

$$\max \vec{c}^T \vec{x} \text{ s.t. } A\vec{x} \le \vec{b}, \vec{x} \ge 0 \quad (P)$$

$$\min \vec{b}^T \vec{y} \text{ s.t. } A^T \vec{y} \ge \vec{c}, y \ge 0 \quad (D)$$

Duality (1)

Let $A \in \mathbb{R}^{m \times n}$, $\vec{c} \in \mathbb{R}^n$, and $\vec{b} \in \mathbb{R}^m$.

Assume there are $\vec{x} \in \mathbb{R}^n$ and $\vec{y} \in \mathbb{R}^m$ such that $\vec{x} \geq \vec{o}$ and $\vec{y} \geq \vec{o}$, $\vec{y}^T A \geq \vec{c}^T$, and $A\vec{x} \leq \vec{b}$.

Observation *

$$\vec{c}^T \vec{x} \leq \vec{y}^T A \vec{x} \leq \vec{y}^T \vec{b}.$$

We rephrase this observation in terms of two LOP's (P) and (D).

Duality (2)

The LOP (P):

Maximize

$$\vec{c}^T \vec{x}$$

subject to

$$A\vec{x} \leq \vec{b} \text{ and } \vec{x} \geq \vec{o}.$$

The LOP (D):

Minimize

$$\vec{y}^T \vec{b}$$

subject to

$$\vec{y}^T A \geq \vec{c}^T \text{ and } \vec{y} \geq \vec{o}.$$

Recall Some Facts

Let A be an (m, n) - Matrix
$\vec{x}$ be an $(n, 1)$ - vector
$\vec{b}$ be an $(m, 1)$ - vector

Remember:

$$A \cdot \vec{x} = \vec{b}$$

$$\Leftrightarrow (A \cdot v)^T = \vec{b}^T$$

$$\Leftrightarrow v^T \cdot A^T = \vec{b}^T$$

We can multiply:

- a row vector and a matrix:
  $(1, m) \cdot (m, n) = (1, n)$
- a matrix and a column vector:
  $(m, n) \cdot (n, 1) = (m, 1)$

Duality (3)

Our initial Observation * now reads as follows.

Theorem 5 (Weak Duality Theorem (WDT))

(i) *If $\vec{x}_0$ and $\vec{y}_0$ are feasible solutions of (P) and (D), then $\vec{c}^T\vec{x} \leq \vec{y}^T\vec{b}$.*

(ii) *If the objective function $\vec{c}^T\vec{x}$ of (P) is unbounded from above, then (D) has no feasible solution.*

(iii) *If the objective function $\vec{y}^T\vec{b}$ of (D) is unbounded from below, then (P) has no feasible solution.*

Remarks:

- (P) is called the *primal LOP*, (D) the *dual LOP*.
- Note that (i) holds especially for optimal feasible solutions of (P) and (D), respectively.
- It is possible that both LOP's, (P) and (D), don't have feasible solutions.

Duality (4)

Theorem 6 (Strong Duality Theorem (SDT))

(1) *If there are feasible solutions of (P) and its objective function* $\vec{c}^T\vec{x}$ *is bounded from above, then (P) and (D) have optimal feasible solutions* $\vec{x}^*$ *and* $\vec{y}^*$*, respectively, and*

$$\vec{c}^T\vec{x}^* = (\vec{y}^*)^T\vec{b}.$$

(2) *If there are feasible solutions of (D) and its objective function* $\vec{y}^T\vec{b}$ *is bounded from below, then (P) and (D) have optimal feasible solutions* $\vec{x}^*$ *and* $\vec{y}^*$*, respectively, and*

$$\vec{c}^T\vec{x}^* = (\vec{y}^*)^T\vec{b}.$$

Duality (5)

We want to give a proof of (1) of Theorem 6 using ideas from the simplex algorithm. The proof is a modification of the one given in the book by Gärtner and Matousek[1].

To this end we recall some of the facts we already know.

(a) If the objective function of (P) $\vec{c}^T\vec{x}$ is bounded on its feasible set $M = \{\vec{x} \in \mathbb{R}^n \mid A\vec{x} \leq \vec{b}, \vec{x} \geq \vec{o}\}$ and $M \neq \emptyset$, then (P) has an optimal solution.

(b) (P) has an optimal solution $\vec{x}^*$ if and only if the standard form LOP ($\hat{\text{P}}$) (see next slide) has one.

(c) If ($\hat{\text{P}}$) has an optimal solution, then it has an optimal basic solution too.

1: J.Matousek and B. Gärtner, *Understanding and Using Linear Programming*, Springer, 2006.

Transforming P

$$(\hat{P})$$

$$\max \quad c^T \vec{x}$$

$$\text{s.t.} \quad A\vec{x} + \vec{z} = \vec{b}$$

$$\Longleftrightarrow$$

$$\max \quad (\vec{c}^{\,T}, \vec{o}^{\,T}) \begin{pmatrix} \vec{x} \\ \vec{z} \end{pmatrix}$$

$$\text{s.t.} \quad \begin{pmatrix} A \mid I \end{pmatrix} \underbrace{\begin{pmatrix} \vec{x} \\ \vec{z} \end{pmatrix}}_{\hat{\vec{x}}} = \vec{b}$$

Duality (6)

The LOP $(\hat{\text{P}})$

Maximize

$$\vec{\hat{c}}^T \vec{\hat{x}}$$

subject to

$$\hat{A}\vec{\hat{x}} = \vec{b}, \ \vec{\hat{x}} \geq \vec{o}$$

with

$$\hat{A} = (A, I), \ \ \vec{\hat{x}} = (\vec{x}, \vec{z})^T, \ \ \vec{\hat{c}} = (\vec{c}, \vec{o})^T.$$

Note that we do not assume that $\vec{b} \geq \vec{o}$.

Duality (7)

Next, we need to reformulate the optimality criterion for basic solutions of $(\hat{\mathrm{P}})$.

Let $\vec{\hat{x}}$ be a feasible basic solution of $(\hat{\mathrm{P}})$.

We let $\hat{A}_B$ be the matrix consisting of the columns of A forming the basis corresponding to $\vec{\hat{x}}$, and $\hat{A}_N$ the matrix consisting of the remaining columns.

Likewise, we write $\vec{h}_B$ and $\vec{h}_N$ for the vectors consisting of the entries of a vector $\vec{h} \in \mathbb{R}^{n+m}$ corresponding to $\hat{A}_B$ and $\hat{A}_N$, respectively.

Thus, $\hat{A}\vec{h} = \hat{A}_B\vec{h}_B + \hat{A}_N\vec{h}_N$.

Let $\vec{\hat{y}}$ be a feasible solution $(\hat{\mathrm{P}})$.

Note that $\hat{A}_B$ is an invertible $m \times m$-matrix.

Duality (8)

Since $\vec{\hat{x}}$ and $\vec{\hat{y}}$ are solutions of $(\hat{\mathrm{P}})$ we know

$$\vec{b} = \hat{A}\vec{\hat{x}} = \hat{A}_B\vec{\hat{x}}_b = \hat{A}_B\vec{\hat{y}}_B + \hat{A}_N\vec{\hat{y}}_N.$$

Hence,

$$\vec{\hat{y}}_B = \vec{\hat{x}}_B - \hat{A}_B^{-1}\hat{A}_N\vec{\hat{y}}_N$$

and

$$\vec{\hat{c}}^T\vec{\hat{y}} = \vec{\hat{c}}_B^T(\vec{\hat{x}}_B - \hat{A}_B^{-1}\hat{A}_N\vec{\hat{y}}_N) + \vec{\hat{c}}_N^T\vec{\hat{y}}_N = \vec{\hat{c}}_B^T\vec{\hat{x}}_B + (\vec{\hat{c}}_N^T - \vec{\hat{c}}_B^T\hat{A}_B^{-1}\hat{A}_N)\vec{\hat{y}}_N.$$

Since $\vec{\hat{c}}^T\vec{\hat{x}} = \vec{\hat{c}}_B^T\vec{\hat{x}}_B$ it follows that $\vec{\hat{x}}$ is optimal if and only if

$$(\vec{\hat{c}}_N^T - \vec{\hat{c}}_B^T\hat{A}_B^{-1}\hat{A}_N)\vec{\hat{y}}_N \le 0$$

for all feasible solutions $\vec{\hat{y}}$ of $(\hat{\mathrm{P}})$. Because $\vec{\hat{y}} \ge \vec{o}$, we finally get that $\vec{\hat{x}}$ is optimal if and only if

$$\vec{\hat{c}}_B^T\hat{A}_B^{-1}\hat{A}_N \ge \vec{\hat{c}}_N^T.$$

Duality (9)

Since $\vec{\hat{c}}_B^T \hat{A}_B^{-1} \hat{A}_B = \vec{\hat{c}}_B^T$ the inequality $\vec{\hat{c}}_B^T \hat{A}_B^{-1} \hat{A}_N \geq \vec{\hat{c}}_N^T$ is equivalent to

$$\vec{\hat{c}}_B^T \hat{A}_B^{-1} \hat{A} = \vec{\hat{c}}_B^T \hat{A}_B^{-1} (\hat{A}_B, \hat{A}_N) = (\vec{\hat{c}}_B^T, \vec{\hat{c}}_B^T \hat{A}_B^{-1} \hat{A}_N) \geq (\vec{\hat{c}}_B^T, \vec{\hat{c}}_N^T) = \vec{\hat{c}}^T.$$

We recall that $\vec{\hat{c}}^T = (\vec{c}^T, \vec{o}^T)$ and $\hat{A} = (A, I)$.

Hence, $\vec{\hat{c}}_B^T \hat{A}_B^{-1} \hat{A} \geq \vec{\hat{c}}^T$ is equivalent to $\vec{\hat{c}}_B^T \hat{A}_B^{-1} A \geq \vec{c}^T$ and $\vec{\hat{c}}_B^T \hat{A}_B^{-1} \geq \vec{o}^T$.

Thus, we have proved the following proposition.

Proposition 1

A feasible basic solution $\vec{\hat{x}}$ of $(\hat{\mathrm{P}})$ is optimal if and only if

$$\vec{\hat{c}}_B^T \hat{A}_B^{-1} A \geq \vec{c}^T \;\; \textit{and} \;\; \vec{\hat{c}}_B^T \hat{A}_B^{-1} \geq \vec{o}^T$$

Duality (10)

Now we are ready to prove assertion (1) of Theorem 6 (SDT).

Proof.

If (P) is feasible and bounded, it follows from (a) that (P) has an optimal solution.

By (b) and (c) it follows that $(\hat{\mathrm{P}})$has an optimal basic solution $\vec{\hat{x}}$.

We let $\vec{u}^T = \vec{\hat{c}}_B^T \hat{A}_B^{-1}$.

By the above proposition $\vec{u}$ is a feasible solution of (D).

Recall that $\vec{\hat{x}}_B = \hat{A}_B^{-1} \vec{b}$.

Hence,

$$\vec{u}^T \vec{b} = \vec{\hat{c}}_B^T \hat{A}_B^{-1} \vec{b} = \vec{\hat{c}}_B^T \vec{\hat{x}}_B = \vec{\hat{c}}^T \vec{\hat{x}} = \vec{c}^T \vec{x}.$$

By the WDT it follows that $\vec{u}$ is an optimal solution of $(\hat{\mathrm{P}})$. $\square$

Duality (11)

The following 'Dualization Recipe' is taken from the book by Gärtner and Matousek[3].

	Primal LOP	Dual LOP
Variables	$x_1, x_2, \ldots, x_n$	$y_1, y_2, \ldots, y_m$
Matrix	A	A^T
Right hand side	$\vec{b}$	$\vec{c}$
Objective function	$\max \vec{c}^T \vec{x}$	$\min \vec{b}^T \vec{y}$
Constraints	ith constraint has $\leq$	$y_i \geq 0$
	ith constraint has $\geq$	$y_i \leq 0$
	ith constraint has $=$	$y_i \in \mathbb{R}$
	$x_j \geq 0$	jth constraint has $\geq$
	$x_j \leq 0$	jth constraint has $\leq$
	$x_j \in \mathbb{R}$	jth constraint has $=$

3: J.Matousek and B. Gärtner, *Understanding and Using Linear Programming*, Springer, 2006.

Branch and Bound – The Basic Algorithm Idea (1)

- **The idea of branch and bound methods**
  - Consider a (hard to solve) optimization problem

    minimize $f(x)$ subject to $x \in M$ (1)

  - Associate a *relaxed* optimization problem

    minimize $g(x)$ subject to $x \in R$ (2)

    such that
    - $M \subseteq R$,
    - If $x \in R$, then $g(x) = f(x)$, and
    - (2) can be solved efficiently.

Branch and Bound – The Basic Algorithm Idea (2)

- Recall:
  - An *optimal solution* for (1) is an element x^* of M such that $f(x) \geq f(x^*)$ for all x in M (likewise for (2)).

- Claim 1:
  a) If y^* is an optimal solution for (2), then $f(x) \geq f(y^*)$ for all x in M.
  b) If y^* is an optimal solution for (2) and $y^* \in M$, then y^* is an optimal solution for (1) too.

- Claim 2:
  - Let $M = M_1 \cup M_2$ be a partition of M, and let x^*_k be the optimal solutions for the OPs

$$\text{minimize } f(x) \quad \text{subject to } x_k \in M_k \qquad (1_k)$$

  with $k \in \{1, 2\}$.
  - Then $\text{argmin}_x \{f(x^*_1), f(x^*_2)\}$ is an optimal solution for (1).

Branch and Bound – The Basic Algorithm Idea (3)

- Let R' be a nonempty subset of R, and consider the optimization problem

$$\text{minimize } g(y) \quad \text{subject to } y \in R' \qquad (R').$$

  - If (R') has a solution, then SOLVE(R') returns a pair $(y^*, g(y^*))$ consisting of an optimal solution y^* of (R') and the corresponding value $g(y^*)$, otherwise SOLVE(R') returns (nil, ∞).

  - If SOLVE(R') yields an optimal solution $y^* \notin M$, then BRANCH(R', y^*) returns two disjoint subsets R'_1, R'_2 of R' such that

$$M \cap R' = (M \cap R'_1) \cup (M \cap R'_2)$$

Branch and Bound – Iterative Implementation

- $\mathbf{L} \leftarrow \{R\}$; $best \leftarrow \infty$; $y_{best} \leftarrow$ nil; // initialization
- while $\mathbf{L} \neq \emptyset$ do

```
begin
   choose B ∈ L;                                   // according to a specific rule
   (y*, g(y*)) ← SOLVE(B);
   if ((y* ∈ M) and (g(y*) < best)) then {
            best ← g(y*);
            y_best ← y*;
            remove B from L;          }
   else {
      if (g(y*) ≥ best) then remove B from L;   // bounding
      else {                                    // branching
            remove B from L;
            (B_1, B_2) ← BRANCH(B, y*);
            L ← L ∪ {B_1, B_2};              } }
end
```


Branch and Bound – Recursive Implementation

- $A \leftarrow M$; $B \leftarrow R$; $best \leftarrow \infty$; $y_{best} \leftarrow$ nil; // initialization
- procedure BB(A,B,f,g)

```
begin
   (y*, g(y*)) ← SOLVE(B);
   if (y* ∈ A) then {
      if (g(y*) < best) then {
         best ← g(y*); y_best ← y*; return; } }
   else {
      if (g(y*) ≥ best) then return;          // bounding
      else {                                  // branching
         (B_1, B_2) ← BRANCH(B, y*);
         BB(A, B_1, f, g);
         BB(A, B_2, f, g);
         return;                         } }
end
```

Branch and Bound – Termination and Complexity

- Remarks:
  - Branch and Bound yields an (exact) optimal solution provided there is a constant *K* (depending on the input) such that all subproblems obtained after at most *K* repetitions of BRANCH either have no optimal solution or their optimal solutions are in *M.*
  - Branch and Bound is not (necessarily) efficient – in the worst case, all possible combinations of integer values for the components x_i of vector x will be tried.

Branch and Bound for Mixed Integer Problems (MIP)

- Consider the MIP (1)

minimize $f(\boldsymbol{x}) = \boldsymbol{c}\,\boldsymbol{x}$

subject to $\boldsymbol{A}\,\boldsymbol{x} = \mathbf{b}$,
$\boldsymbol{x} \geq \boldsymbol{0}$, and
x_j is integer for all $j \in I$

where $\boldsymbol{x} = (x_{1,\dots,}x_n)$, $\boldsymbol{b} \geq \boldsymbol{0}$, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ has rank $m \leq n$, and $I \subseteq \{1, \dots, n\}$

- Choose as relaxed problem the LOP (2) (called *LP-relaxation*)

minimize $f(\boldsymbol{x}) = \boldsymbol{c}\,\boldsymbol{x}$

subject to $\boldsymbol{A}\,\boldsymbol{x} = \mathbf{b}$,
$\boldsymbol{x} \geq \boldsymbol{0}$

where $\boldsymbol{x} = (x_1, \dots, x_n)$, $\boldsymbol{b} \geq \boldsymbol{0}$, and $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ has rank $m \leq n$

Branch and Bound for MIP

- SOLVE can be any method to solve LOPs (e.g. the simplex algortihm)
- If $(y^*, g(y^*))$ is an optimal solution where y^*_k is not an integer (with k being the smallest such index), then BRANCH(B, y^*) augments new inequalitities to B, i.e.

BRANCH(B, y^*) = (B_1, B_2)

where

$$B_1 = \{y \in B \mid y_k \leq \lfloor y^*_k \rfloor\},$$
$$B_2 = \{y \in B \mid y_k \geq \lceil y^*_k \rceil\}.$$

Notation: $\lfloor y^*_k \rfloor$ is the greatest integer not greater than y^*_k and
$\lceil y^*_k \rceil$ is the smallest integer not smaller than y^*_k

Cutting planes for MIP

Consider:

- The MIP (1)

minimize	$f(\mathbf{x}) = \mathbf{c}\,\mathbf{x}$
subject to	$\mathbf{x} \in M = \{\,\mathbf{x} \mid \mathbf{A}\,\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}$, and x_j is integer for all $j \in I\,\}$
where	$\mathbf{x} = (x_1, \ldots, x_n)$ and $I \subseteq \{1, \ldots, n\}$

- Its LP-relaxation (2)

minimize	$f(\mathbf{x}) = \mathbf{c}\,\mathbf{x}$
subject to	$\mathbf{x} \in R = \{\mathbf{x} \mid \mathbf{A}\,\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\,\}$
where	$\mathbf{x} = (x_1, \ldots, x_n)$

Cutting planes for MIP

- A *valid cut* for (1) is an equality $\boldsymbol{d}\,\boldsymbol{x} \leq q$ such that
  - $\{\boldsymbol{x} \mid \boldsymbol{A}\,\boldsymbol{x} = \mathbf{b}, \boldsymbol{x} \geq \boldsymbol{0}, \boldsymbol{d}\,\boldsymbol{x} \leq q\} \neq R$ and
  - $\{\boldsymbol{x} \mid \boldsymbol{A}\,\boldsymbol{x} = \mathbf{b}, \boldsymbol{x} \geq \boldsymbol{0}$, and x_j is integer for all $j \in I$, $\boldsymbol{d}\,\boldsymbol{x} \leq \mathrm{q}\} = M.$

- Outline of a cutting plane algorithm

```
begin
    B ← R;
    (y*, g(y*)) ← SOLVE(B);
    while (y* ∉ M) do
    begin
        compute a valid cut d x ≤ q for B;
        B ← { x ∈ B | d x ≤ q };
        (y*, g(y*)) ← SOLVE(B);
    end;
end
```


Cutting planes for MIP

- There are two kinds of cuts:
  - problem specific ones (e.g. for the knap sack problem), and
  - general purpose ones (Gomory cuts).

- Gomory cuts (here explained for $I = \{1, \ldots, n\}$)
  - Let $\boldsymbol{x} = (x_{1,\ldots,} x_n)$ be a basic solution of the LP-relaxation such that x_i is not an integer (i smallest index if multiple such indices exist).
  - Then x_i is a basic variable and from the simplex tableau we know that

$$x_i + \sum a_{ij}\, x_j = b_i \qquad (a)$$

  - Equation (a) implies

$$x_i + \sum \lfloor a_{ij} \rfloor x_j - \lfloor b_i \rfloor = b_i - \lfloor b_i \rfloor - \sum (a_{ij} - \lfloor a_{ij} \rfloor)\, x_j \qquad (b)$$

Note that here a_{ij} and b_i are the values in the tableau after the computation of $\boldsymbol{x}$, not the values of the original matrix $\boldsymbol{A}$ or vector $\boldsymbol{b}$

Cutting planes for MIP

- For any point $\boldsymbol{x} \in \boldsymbol{M}$ the right hand side of (b) is less than 1, and the left hand side is an integer.
- Consequently,

$$b_i - \lfloor b_i \rfloor - \sum (a_{ij} - \lfloor a_{ij} \rfloor) x_j \leq 0 \qquad \text{(c)}$$

  for any point $\boldsymbol{x} \in \boldsymbol{M}$
- Furthermore, for the original basic solution $\boldsymbol{x} = (x_1, \ldots, x_n)$ it holds that

$$b_i - \lfloor b_i \rfloor - \sum (a_{ij} - \lfloor a_{ij} \rfloor) x_j = b_i - \lfloor b_i \rfloor$$

  (because all non basic variables $x_j = 0$), and it is not an integer (because x_i and so also b_i is not an integer), and therefore

$$b_i - \lfloor b_i \rfloor - \sum (a_{ij} - \lfloor a_{ij} \rfloor) x_j = b_i - \lfloor b_i \rfloor > 0$$

  i.e. (c) excludes $\boldsymbol{x}$
- Hence, the inequality (c) is a valid cut, and it is called a Gomory Cut.
- Note: to yield standard form again, a new slack variable x_{n+1} will be introduced

$$- x_{n+1} + \sum (a_{ij} - \lfloor a_{ij} \rfloor) x_j = b_i - \lfloor b_i \rfloor$$

  x_{n+1} is integer

Conclusions

- Linear Optimization Problems (LOPs) consist of
  - finding an optimal (minimal or maximal, respectively) solution to a linear objective function under a set of linear constraints for their variables
- The simplex algorithm can be seen as a clever application of repeated Gauss-Jordan transformation, used to solve LOPs
  - In most practical applications, its time complexity is linear in m = # of constraints and n = # of variables
  - In the worst case, it can lead to exponentially growing running times
  - There are other algorithms that can guarantee polynomial time complexity
- (Linear) Mixed Integer Problems
  - have additional integer constraints for some of their variables x_j, and
  - can thus not directly be solved with a solver algorithm for LOPs
- Methods like “Branch and Bound” as well as “Cutting Planes”
  - perform repeated iterations of a solver algorithm on relaxed problems (= without integer constraints),
  - while adding additional constraints for cutting out solutions that violate the integer constraints,
  - until they find an optimal solution that fulfills all integer constraints