Contents

1 Linear Programming **2**

1.1 Standard Forms of Linear Programs 2
1.2 Geometric Interpretation of Linear Programs and the Simplex Algorithm 3
1.2.1 Algebraic Formulation of the Simplex Algorithm 4
1.2.2 Termination 11
1.2.3 Finding a First Feasible Basis: Phase I of the Simplex Method 14
1.3 Min Cost Network Flow 14
1.4 Linear Programming Duality 22
1.4.1 Algebraic Motivation of Duality 22
1.4.2 Rules for Forming the Dual 23
1.4.3 Strong Duality Theorem and Complementary Slackness 25
1.4.4 Geometric Interpretation of Duality I 28
1.4.5 Geometric Interpretation of Duality II (this may be skipped) 29
1.5 The Dual Simplex Method 31
1.6 Sensitivity Analysis 33
1.7 Reoptimization 33
1.8 Totally Unimodular Constraint Matrices 34
1.9 Primal-Dual Combinatorial Algorithms 36
1.10 Running Time of Simplex Method 36
1.11 Alternative Algorithms for Solving Linear Programs 36

Chapter 1

Linear Programming

1.1 Standard Forms of Linear Programs

A linear program asks for the maximum (or minimum) value of a linear function $c^T x$ in n variables $x_1, \ldots, x_n$ subject to a set of m linear *constraints (Nebenbedingung)*. The linear function $c^T x$ is called the *objective function (Zielfunktion)*. A linear constraint is either an equality $a^T x = b$ or an inequality $a^T x \leq b$ or $a^T x \geq b$; strict inequalities $a^T x < b$ or $a^T x > b$ are not permitted. Variables either range over all real numbers (unconstrained variables) or over the non-negative real numbers (non-negative variables) or over the non-positive numbers (non-positive variables).

In the formulations above x, c, and a are column vectors of size n. We use the superscript T to denote transposition. b is a scalar value.

In our examples, we use equalities and inequalities and constrained and unconstrained variables. The objective is the elegance of the LP. In our theoretical treatment, we restrict consideration to LPs in standard form. The goal is elegance and simplicity of the mathematics and the algorithms.

Different standard forms are convenient for different parts of the theory. The following rules can be used to transform LPs.

- A maximization problem can be turned into a minimization problem and vice versa by changing the sign of the objective function, i.e., by replacing $c^T x$ by $-c^T x$.
- An equation $a^T x = b$ can be replaced by inequalities $a^T x \leq b$ and $a^T x \geq b$.
- An inequality $a^T x \geq b$ may be replaced by $(-a)^T x \leq -b$.
- An inequality $a^T x \leq b$ may be replaced by an equality $a^T x + s = b$, where s is a new non-negative variable. s is called a slack variable (Schlupfvariable).
- An non-negative variable x may be replaced by an unconstrained variable x and the constraint $x \geq 0$.
- An unconstrained variable x may be replaced by the difference $x' - x''$ of two new non-negative variables x' and x''.

A frequently used standard form is

$$\text{maximize} \quad c^T x \quad \text{subject to} \quad Ax \leq b, \quad x \in \mathbb{R}^n_{\geq 0} .$$

We have n non-negative variables $x_1, \dots x_n$. The m linear constraints are summarized into $Ax \le b$; A is an $m \times n$ matrix, the constraint matrix, and b is a column vector of size m, the right hand side. Observe that we use b for the right hand side of a single constraint and also for the vector of right hand sides. If we want to emphasize the distinction, we write b_i for the right hand side of the i-th constraint.

We need to comment on the use of the comparison operator between vectors. If x and y are vectors of the same dimension, $x \le y$ holds if x is component-wise bounded by y. Hence $\neg(x \le y)$ or $x \not\le y$ means that there is some component in which x is larger than y. Thus $x \not\le y$ is not the same as $x > y$. The comparison operator for vectors interacts nicely with scalar products between vectors. In particular,

- if $c \ge 0$ and $x \ge 0$, $c^T x \ge 0$
- if $c \le 0$ and $x \ge 0$, $c^T x \le 0$
- if $c \le 0$ and $x \le 0$, $c^T x \ge 0$

A point $x^* \in \mathbb{R}^n$ is called *feasible* or *feasible solution* if it satisfies all constraints, i.e., $Ax^* \le b$ and $x^* \in \mathbb{R}^n_{\ge 0}$. It is infeasible otherwise. An LP is feasible, if there is at least one feasible solution, and it is infeasible otherwise. A point x^* is an optimal solution if it feasible and if $c^T x^* \ge c^T x'$ for all feasible x'.

Alternative standard forms are

$$\text{maximize} \quad c^T x \quad \text{subject to} \quad Ax \le b, \quad x \in \mathbb{R}^n$$

or

$$\text{maximize} \quad c^T x \quad \text{subject to} \quad Ax = b, \quad x \in \mathbb{R}^n_{\ge 0}\ .$$

1.2 Geometric Interpretation of Linear Programs and the Simplex Algorithm

Consider the following LP.

$$\text{maximize} \quad c^T x \quad \text{subject to} \quad Ax \le b$$

Our variables are unconstrained. Every constraint $a_i^T x \le b_i$ defines a closed half-space. The boundary of the half-space is the plane $a_i^T x = b_i$ and the half-space consists of all points on the plane and on one side of the plane. The feasible region is the intersection of m closed half-spaces. The feasible region is closed convex set which is bounded by planes. Such a region is called a polytope. The LP is infeasible iff the feasible region is empty. If the feasible region is non-empty it may be either bounded or unbounded. In the former case the feasible region is a polyhedron.

TODO: Illustrate the concepts by drawings in the plane.

Assume that the feasible region is non-empty and use P to denote the feasible region. The goal is to find the feasible point that is maximal in the direction c. There are two cases:

- Either we have feasible points x with arbitrarily large values of $c^T x$. We say that the LP is unbounded. In this case the feasible region must be unbounded in the direction c, i.e., there is no real number z such that $P = P \cap (c^T x \leq z)$.

- The feasible region is bounded in the direction c. We say that the LP is bounded (this does not imply that P is bounded, e.g., P may be unbounded in the direction $-c$). In this case there is a smallest number z^* such that $P = P \cap (c^T x \leq z^*)$; z^* is the optimal objective value. The feasible region P is contained in the half-space $c^T x \leq z^*$ and there is at least one feasible point x^* lying in the plane $c^T x = z^*$. The plane is tangent to the feasible region. If there is exactly one feasible point lying in the plane $c^T x = z^*$, the optimal solution is unique and is a vertex of P. If more than one feasible point are contained in the plane, the optimal solution is not unique.

Let us summarize.

Theorem 1 *An LP is either feasible or infeasible. A feasible LP is either bounded or unbounded. A bounded LP has an optimal solution.*

Our geometric interpretation also suggests an algorithm for finding an optimal solution by walking along the edges of the polytope P. This is the famous simplex algorithm invented by Dantzig in 1951.

1. determine an arbitrary vertex v of the feasible region P
2. consider the edges of P incident to v. If walking along any one of them does not improve the objective function, stop: v is an optimal solution.
3. otherwise, select one the improving edges e and walk along the edge. If the edge is unbounded, the LP is unbounded. Stop.
4. otherwise replace v by the other endpoint of e and go back to step 2.

1.2.1 Algebraic Formulation of the Simplex Algorithm

We use the following standard form for this section.

$$\text{maximize} \quad c^T x \quad \text{subject to} \quad Ax = b, \quad x \in \mathbb{R}^n_{\geq 0} .$$

We assume that the rows of A are linearly independent (otherwise, we may simply drop one of the constraints) and hence $\text{rank}(A) = m \leq n$. Recall that the standard form used in the preceding section is easily converted into the standard form used in this section by adding a slack variable to each constraint.

Example: Throughout this section we use the following simple example. We have non-negative variables x and y and the feasible region is defined by the inequalities $x \geq 1$, $x \leq 2$, $y - x \geq 0$. , see Figure 1.1. We add a non-negative slack variable for each constraint and obtain $x - s = 1$, $x + t = 2$, $y - x - u = 0$. Our feasible region becomes a subset of $\mathbb{R}^5_{\geq 0}$:

$$P = \{(x, y, s, t, u)\ ;\ x - s = 1, \quad x + t = 2, \quad y - x - u = 0, \quad x, y, s, t, u \in \mathbb{R}_{\geq 0}\} .$$

TODO

Figure 1.1: The feasible region P_0 is unbounded. The boundary of the feasible region consists of vertices $v = (1,1)$ and $w = (2,2)$ and three edges e, f, and g. The edges e and g are unbounded and the edge $f = vw$ is bounded. We will consider four objective functions: (1) For $z = x - y/2$, w is the unique optimal solution. The maximal objective value is 1. (2) For $z = x - y$, all points on f are optimal. The maximal objective value is 0. (3) For $z = -x - y$, v is the unique optimal solution. The maximal objective value is -2. (4) For $z = y$, the LP is unbounded.

There is a one-to-one correspondence between the points in P_0 and the points in P. For example, $(2,2)$ corresponds to $(2,2,1,0,0)$. It is easy to find this vertex in our system of constraints. We simply solve the system for the variables x, y, and s.

$$\begin{aligned} x &= 2 - t \\ y &= 2 - t + u \\ s &= 1 - t \end{aligned}$$

Generally, we obtain vertices by solving our system of constraints for a subset of m variables. The appropriate notion is the notion of a basis.

A *basis* is a $m \times m$ invertible submatrix of A. We denote such submatrices as follows. Let A_i denote the i-th column of A and let $B = \{B_1, B_2, \ldots, B_m\}$ be a set of m column indices, i.e., a subset of $\{1, \ldots, n\}$ of size m. We use A_B to denote the $m \times m$ submatrix of A formed by the columns A_{B_i}, $1 \le i \le m$. B and also A_B is called a *basis* if A_B is invertible. The variables selected by the indices in B are denoted x_B and are called the basic variables.

We use $N = \{1, \ldots, n\} \setminus B$ to denote the indices not in B, and A_N for the submatrix selected by the indices in N and x_N for the variables selected by the indices in N. The number of non-basic variables is $n - m$.

We can now rewrite $Ax = b$ as

$$A_B x_B + A_N x_N = b \ .$$

In the basic solution corresponding to basis B we set the non-basic variables to 0 and then solve the equation above for the basic variables, i.e.,

$$x_B = A_B^{-1} b \quad \text{and} \quad x_N = 0 \ .$$

We call $(A_B^{-1} b, 0)$ the basic solution corresponding to B.

B is a feasible basis if the basic solution corresponding to B is feasible, i.e., if $A_B^{-1} b \ge 0$. A basis is optimal if it is feasible and the solution corresponding to it is optimal

We postpone the question of how to find a basic feasible solution and proceed under the assumption that we have a feasible basic B. The objective function value for the basic solution corresponding to B is

$$z = c_B^T x_B + c_N^T x_N = c_B^T A_B^{-1} b \ .$$

How can we determine whether B is an optimal basis? In the geometric view we argued. If there is no improving edge incident to a vertex, the vertex is optimal. Edges correspond to non-basic variables. We move by making a single non-basic variable positive. Which one? In order to answer this question, we view the basic variables as dependent variables and the non-basic variables as independent variables, i.e., we write

$$x_B = A_B^{-1}b - A_B^{-1}A_N x_N \ .$$

This is the same as solving our constraint system for the basic variables. The solved form $x_B = A_B^{-1}b - A_B^{-1}A_N x_N$ is called a *dictionary*. Plugging into the objective function yields

$$z = c_B^T x_B + c_N^T x_N = c_B^T(A_B^{-1}b - A_B^{-1}A_N x_N) + c_N^T X_N = c_B^T A_B^{-1}b + (c_N^T - c_B^T A_B^{-1}A_N)x_N$$

and tells us how the objective value changes as the result of a change in the non-basic variables. The objective value is a linear function in the non-basic variables.

We call the vector $c_N^T - c_B^T A_B^{-1}A_N$ the vector of *reduced costs* or *prizes* of the non-basic variables.

Theorem 2 *If the reduced costs are non-positive the basis is optimal.*

$$c_N^T - c_B^T A_B^{-1}A_N \le 0 \quad \textit{implies} \quad B \textit{ is an optimal} \ .$$

Proof: Consider any feasible solution $x' = (x'_B, x'_N)$. Then $x'_B = A_B^{-1}b - A_B^{-1}A_N x'_N$ and $x'_N \ge 0$. Thus

$$c^T x' = c_B^T x'_B + c_N^T x'_N = c_B^T A_B^{-1}b + (c_N^T - c_B^T A_B^{-1}A_N)x'_N \le c_B^T A_B^{-1}b$$

where the last inequality follows from the fact that $c_N^T - c_B^T A_B^{-1}A_N \le 0$ and $x'_N \ge 0$. ∎

Theorem 2 is the termination criterion of the Simplex algorithm. Stop when all reduced costs are non-positive . In this situation, the current solution is optimal.

Example, continued: Above, we solved our system for our basis variables x, y, and s.

$$\begin{aligned} x &= 2 - t \\ y &= 2 - t + u \\ s &= 1 - t \end{aligned}$$

Let us look at the objective value as a function of the non-basic variables t and u.

- For $z = x - y/2$ we obtain $z = (2-t) - (2-t+u)/2 = 1 - t/2 - u/2$ and hence our basis is optimal. In fact, it is the unique optimum basis since increasing any of the non-basic variables to a positive value will strictly decrease the objective value.
- For $z = x - y$ we obtain $z = (2-t) - (2-t+u) = 0 + 0t - u$ and hence our basis is optimal. It is not the unique optimum since increasing the value of t will leave the value of the objective function unchanged.
- For $z = -x - y$ we obtain $z = -(2-t) - (2-t+u) = -4 + 2t - u$ and hence our basis is not optimal. It is profitable to increase the value of t.
- For $z = y$ we obtain $z = 2 - t + u$ and hence our basis is not optimal. It is profitable to increase the value u.

So let us assume that some reduced cost is positive. We use $\overline{c}_N = c_N^T - c_B^T A_B^{-1} A_N$ to denote the vector of reduced costs. It is a vector with one entry for each non-basic variable. Let x_j be a non-basic variable with positive reduced cost. In an implementation of the Simplex algorithm one typically chooses x_j as the variable with the most positive reduced cost.

Increasing the value of x_j to a positive value will increase the objective value. It will also change the values of the basic variables. Recall that we view the basic variables as dependent variables. The dependency of the basic variables on x_j is given by

$$x_B = A_B^{-1} b - A_B^{-1} A_j x_j \ .$$

We want to increase x_j because this will increase the objective value and we want to stay feasible. We therefore increase x_j until one of the basic variables becomes zero. This corresponds to walking along an edge of the polytope. Two cases can arise.

- We can walk forever and the LP is unbounded.
- We can walk a finite amount until we reach the next vertex (= basis).

How can we distinguish the cases algebraically. We need some notation. Let $\overline{b} = A_B^{-1} b$ be the current values of the basic variables and let $\overline{A}_j = A_B^{-1} A_j$ be the vector that tells us how the basic variables depend on x_j.

Theorem 3 *Let x_j be a non-basic variable with positive reduced cost. If $\overline{A}_j$ is a non-negative vector, the LP is unbounded.*

$$\overline{A}_j \geq 0 \quad \textit{implies} \quad \textit{LP is unbounded} \ .$$

Proof: We may increase x_j arbitrarily and stay feasible. This will increase the objective value arbitrarily. ∎

Example, continued: Above, we solved our system for our basis variables x, y, and s.

$$\begin{aligned} x &= 2 - t \\ y &= 2 - t + u \\ s &= 1 - t \end{aligned}$$

For $z = y$ we obtain $z = 2 - t + u$ and hence our basis is not optimal. It is profitable to increase the value u. The value of no basic variable is increased when u is increased and hence the LP is unbounded.

So assume that $\overline{A}_j$ has negative entries. Let x_k be any basic variable and let $\overline{a}_{kj}$ be the entry of $\overline{A}_j$ corresponding to x_k. Then

$$x_k = \overline{b}_k + \overline{a}_{kj} x_j \ .$$

If $\overline{a}_{kj} < 0$, we cannot increase x_j beyond $\overline{b}_k / |\overline{a}_{kj}|$ without destroying feasibility. We want to increase x_j to the maximal possible value and therefore increase it to

$$\min \left\{ \frac{\overline{b}_k}{|\overline{a}_{kj}|} \ ; k \in B \text{ and } \overline{a}_{kj} < 0 \right\} \ .$$

What is our new basis? Consider any of the basic variables x_i which become zero, i.e.,

$$i \in B, \quad \overline{a}_{ij} < 0 \quad \text{and} \quad \frac{\overline{b}_i}{|\overline{a}_{ij}|} = \min \left\{ \frac{\overline{b}_k}{|\overline{a}_{kj}|} \text{ ; } k \in B \text{ and } \overline{a}_{kj} < 0 \right\} .$$

There might be several such i. In this case we chose one arbitrarily. We remove x_i from the basis and replace it by x_j. Our new basis is $B' = B \setminus i \cup j$. We still need to argue that the submatrix $A_{B'}$ is invertible and that the basis is feasible (the latter should be obvious by our choice of the new value for x_j).

Theorem 4 *Let x_j be a non-basic variable with positive reduced cost and assume that $\overline{A}_j$ has at least one negative component. Let*

$$i \in B, \quad \overline{a}_{ij} < 0 \quad \textit{and} \quad \frac{\overline{b}_i}{|\overline{a}_{ij}|} = \min \left\{ \frac{\overline{b}_k}{|\overline{a}_{kj}|} \text{ ; } k \in B \textit{ and } \overline{a}_{kj} < 0 \right\} .$$

Then $B' = B \setminus i \cup j$ is a feasible basis. The objective value at B' is at least as large as the objective value at B. If $\overline{b}_i > 0$, the objective value at B' is larger.

Proof: We first show that B' is a basis. We solve the i-th equation for x_j (we can do so because $\overline{a}_{ij} \neq 0$)

$$x_j = (x_i - \overline{b}_i - \sum_{l \in N \setminus j} \overline{a}_{il} x_l)/\overline{a}_{ij}$$

and substitute this expression into the other equations. So for $k \in B \setminus i$, we obtain

$$\begin{aligned} x_k &= \overline{b}_k + \sum_{l \in N} \overline{a}_{kl} x_l \\ &= \overline{b}_k + \sum_{l \in N \setminus j} \overline{a}_{kl} x_l + \overline{a}_{kj}(x_i - \overline{b}_i - \sum_{l \in N \setminus j} \overline{a}_{il} x_l)/\overline{a}_{ij} \\ &= \overline{b}_k - \overline{a}_{kj}\overline{b}_i/\overline{a}_{ij} + \frac{\overline{a}_{kj}}{\overline{a}_{ij}} x_i + \sum_{l \in N \setminus j} (\overline{a}_{kl} - \frac{\overline{a}_{kj}\overline{a}_{il}}{\overline{a}_{ij}}) x_l \end{aligned}$$

We have now solved our constraints for the variables in B' and hence the submatrix $A_{B'}$ is invertible (otherwise we could not solve the equations for the variables in B').

We next argue feasibility:

- $x_j = -\overline{b}_i/\overline{a}_{ij}$ is non-negative, since $\overline{b}_i \geq 0$ and $\overline{a}_{ij} < 0$.
- So consider $k \in B \setminus i$. We have $x_k = \overline{b}_k - \overline{a}_{kj}\overline{b}_i/\overline{a}_{ij}$ where $\overline{b}_k \geq 0$, $\overline{b}_i \geq 0$, and $\overline{a}_{ij} < 0$.
  - If $\overline{a}_{kj} \geq 0$, the new value for x_k is certainly non-negative.
  - If $\overline{a}_{kj} < 0$, the new value for x_k is non-negative since $\overline{b}_k/|\overline{a}_{kj}| \geq \overline{b}_i/|\overline{a}_{ij}|$ by our choice of i.

The new objective value is (let $\overline{z} = c_B^T A_B^{-1} b$ be the current objective value)

$$\begin{aligned} z &= \overline{z} + \sum_{l \in N} \overline{c}_l x_l \\ &= \overline{z} + \sum_{l \in N \setminus j} \overline{c}_l x_l + \overline{c}_j(x_i - \overline{b}_i - \sum_{l \in N \setminus j} \overline{a}_{il} x_l)/\overline{a}_{ij} \\ &= \overline{z} - \frac{\overline{c}_j \overline{b}_i}{\overline{a}_{il}} + \overline{c}_j x_i + \sum_{l \in N \setminus j} (\overline{c}_l - \frac{\overline{c}_j \overline{a}_{ij}}{\overline{a}_{ij}}) x_l \end{aligned}$$

Observe that the objective value does not decrease (that was the point of the exercise) and properly increases if $\overline{b}_i > 0$. ∎

Example, continued: Above, we solved our system for our basis variables x, y, and s.

$$\begin{aligned} x &= 2-t \\ y &= 2-t+u \\ s &= 1-t \end{aligned}$$

For $z = -x-y$ we obtain $z = -(2-t)-(2-t+u) = -4+2t-u$ and hence our basis is not optimal. It is profitable to increase the value of t. The values of all basic variables decrease, when t in increased. We have the following restrictions on t: $x = 2-t$ and $y = 2-t+u$ yield $t \leq 2$ and $s = 1-t$ yields $t \leq 1$. We therefore remove s from the basis and add t to the basis. We obtain the following dictionary.

$$\begin{aligned} x &= 1+s \\ y &= 1+s+u \\ t &= 1-s \end{aligned}$$

The objective function becomes $z = -x-y = -2-2s-u$ and hence the new basis is optimal. In fact, it the unique optimal basis.

Summary of Phase II of the Primal Simplex Algorithm: We now have all the ingredients for phase II of the (primal) simplex method[1]. The simplex method is an iterative method. At the beginning of an iteration we have a feasible basis B, have solved our constraints for the basic variables, and have expressed the objective value as a linear function in the non-basic variables (by substituting the expressions for the basic variables into it). Thus

$$\begin{aligned} x_i &= \overline{b}_i + \sum_{j\in N} \overline{a}_{ij} x_j \qquad \text{for } i \in B \\ z &= \overline{z} + \sum_{j\in N} \overline{c}_j x_j \end{aligned}$$

with $\overline{b}_i \geq 0$.

- If $\overline{c}_j \leq 0$ for all $j \in N$, the current basis is optimal. $x_i = \overline{b}_i$ for $i \in B$ and $x_j = 0$ for $j \in N$ is an optimal solution and $\overline{z}$ is the optimal objective value.
- Otherwise, let $j \in N$ be arbitrary with $\overline{c}_j > 0$.
  - If $\overline{a}_{ij} \geq 0$ for all $i \in B$, the LP is unbounded. All points on the ray $x_i = \overline{b}_i + \overline{a}_{ij}\lambda$ for $i \in B$, $x_j = \lambda$, $\lambda \geq 0$ and $x_l = 0$ for $l \in N \setminus j$ are feasible. The objective value $\overline{z} + \overline{c}_j\lambda$ becomes arbitrarily large for points on the ray.

[1] There is also a dual simplex method and there is also a phase I of the simplex method (concerned with finding an initial feasible basis). We will discuss them later.

– Otherwise, let $i \in B$ be such that $\overline{a}_{ij} < 0$ and

$$\frac{\overline{b}_i}{|\overline{a}_{ij}|} = \min\left\{ \frac{\overline{b}_k}{|\overline{a}_{kj}|} \text{ ; } k \in B \text{ and } \overline{a}_{kj} < 0 \right\} .$$

We change to the basis $B' = B \setminus i \cup j$. The new dictionary is

$$\begin{aligned}
x_j &= \frac{-\overline{b}_i}{\overline{a}_{ij}} && + \frac{1}{\overline{a}_{ij}} x_i + \sum_{l \in N \setminus j} \frac{-\overline{a}_{il}}{\overline{a}_{ij}} x_l \\
x_k &= \overline{b}_k - \overline{a}_{kj}\overline{b}_i/\overline{a}_{ij} && + \frac{\overline{a}_{kj}}{\overline{a}_{ij}} x_i + \sum_{l \in N \setminus j} (\overline{a}_{kl} - \frac{\overline{a}_{kj}\overline{a}_{il}}{\overline{a}_{ij}}) x_l \quad \text{for } k \in B \setminus i
\end{aligned}$$

and the objective function expressed in the new basis becomes

$$z = \overline{z} - \frac{\overline{c}_j \overline{b}_i}{\overline{a}_{ij}} + \overline{c}_j x_i + \sum_{l \in N \setminus j} (\overline{c}_l - \frac{\overline{c}_j \overline{a}_{il}}{\overline{a}_{ij}}) x_l$$

1.2.2 Termination

We have seen in the preceding section that the objective value is monotonically increasing if $\overline{b}_i > 0$ in every iteration of the simplex method where $i \in B$ is the index leaving the basis. This situation deserves a special name.

An LP ($Ax = b$, $x \in \mathbb{R}^n_{\geq 0}$) is *non-degenerate* if all entries of $A_B^{-1}b$ are non-zero for every basis B. It is *degenerate* otherwise. A basis B is called *non-degenerate* if all entries of $A_B^{-1}b$ are non-zero. A basis is *degenerate* otherwise.

Theorem 5 *Phase II of the simplex algorithm terminates when applied to a non-degenerate LP.*

Proof: If all feasible bases are non-degenerate, every iteration increases the value of the objective function and hence no basis can reoccur in the execution.

A basis is defined by a subset of m columns of the constraint matrix and hence the number of bases is bounded by $\binom{n}{m}$. Since no basis can reoccur, the number of iterations is bounded by the same number. ▮

Theorem 5 gives a gigantic bound on the number of iterations in the worst case. Unfortunately, there are examples which actually require an exponential number of iterations. Fortunately, the average number of iterations is much smaller and usually linear. We will come back to this point in Section 1.10.

We discuss two methods of dealing with degenerate LPs: Bland's anti-cycling rule and perturbation. Bland's rule is particularly simple.

Bland's rule: Choose entering and leaving index minimal among the candidate indices. More precisely:

Choose the entering index j minimal among all indices with positive reduced cost and choose the leaving index i minimal among all indices with $i \in B$, $\overline{a}_{ij} < 0$ and

$$\frac{\overline{b}_i}{|\overline{a}_{ij}|} = \min\left\{ \frac{\overline{b}_k}{|\overline{a}_{kj}|} \text{ ; } k \in B \text{ and } \overline{a}_{kj} < 0 \right\} .$$

Theorem 6 *The simplex algorithm terminates when Bland's selection rule is used.*

Proof: See TODO ▮

Bland's rule is used in most implementations of the simplex algorithm. The second technique for dealing with degeneracy is perturbation. Perturbation is a method of wide applicability and therefore I want to discuss it.

It is easy to motivate perturbation geometrically for LPs in the inequality standard form

$$Ax \leq b .$$

The feasible region R is a polytope obtained as the intersection of m closed half-spaces. In n-dimensional space n hyperplanes intersect in a point (in the plane two lines intersect in a point) and degeneracy corresponds to the situation where $n' > n$ of the bounding hyperplanes intersect in a point. This point can be represented as the intersection of any n of the n' hyperplanes and

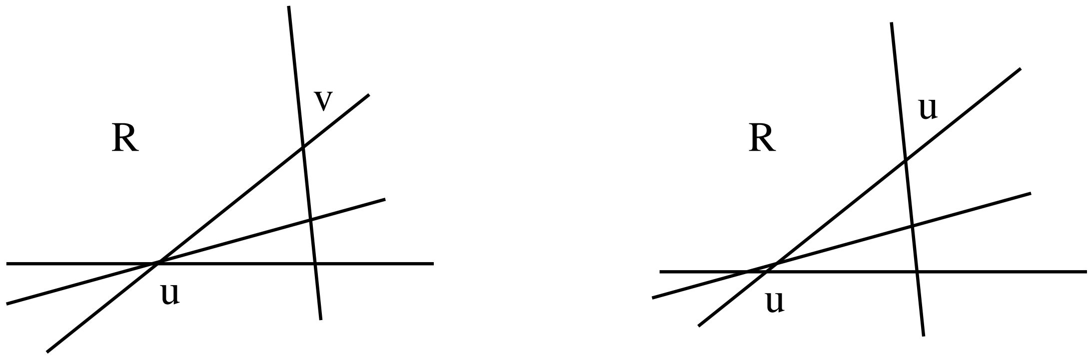

Figure 1.2: In the figure on the left three lines pass through the vertex u of the feasible region R. In the figure on the right we perturbed the problem slightly by moving each lines parallel to itself (algebraically this amounts to a change of the right-hand side. "Globally", the feasible region does not change. "Locally", the vertex u is replaced by two vertices.

this is where the algebraic formulation of the simplex algorithm can get confused. It does not have an explicit representation of the feasible region (it would be much too costly to compute it) and hence there is the danger that is cycles through the different representations of the same vertex.

There is a simple remedy. We simply move each one of the bounding hyperplanes parallel to itself by a tiny amount. This will not change the global structure of the polytope if the perturbation is small enough and it will replace vertices defined by more than n hyperplanes by small non-degenerate polyhedral structures, see Figure 1.2.

We come to the algebraic treatment of perturbation. We change (perturb) the right hand side of our constraint system slightly. More precisely,

$$\text{we replace} \quad b \quad \text{by} \quad b+\vec{\varepsilon}, \quad \text{where} \quad \vec{\varepsilon} = \begin{pmatrix} \varepsilon \\ \varepsilon^2 \\ . \\ . \\ \varepsilon^m \end{pmatrix}$$

and ε is a small positive quantity. We use LP(ε) to denote the LP defined by $Ax = b + \vec{\varepsilon}$. Two remarks are in order.

- ε is small and hence the problem is only changed slightly.
- We change the right-hand sides by vastly different amounts. Observe that $\varepsilon^m \ll \varepsilon^{m-1} \ll \ldots \ll \varepsilon^1$. It would not work to change all right hand sides by the same amount.

We show:

Theorem 7 *There is a* $\delta > 0$ *such that for all* ε *with* $0 < \varepsilon < \delta$*:*

- *LP(ε) is non-degenerate, i.e., we have achieved our goal of obtaining a non-degenerate LP.*
- *Every optimal basis of LP(ε) is an optimal basis of LP, i.e., solving LP(ε) also solves LP.*

- *We can execute the simplex algorithm on LP(ε) (without knowing δ and ε !!!).*

Proof: We address the items in turn.

Let B be a basis. In the basic feasible solution with respect to B we have

$$x_B = A_B^{-1}(b+\vec{\varepsilon}) = A_B^{-1}b + A_B^{-1}\vec{\varepsilon} = \overline{b} + A_B^{-1}\vec{\varepsilon}\ .$$

Consider a particular basic variable x_i and let a_i be the row of A_B^{-1} corresponding to x_i. Then

$$x_i = \overline{b}_i + \sum_{1\le j\le m} a_{ij}\varepsilon^j\ ,$$

i.e., x_i is equal to an m-degree polynomial $p_{B,i}$ evaluated at ε. Since A_B and hence A_B^{-1} is non-singular, we have $a_i \neq 0$, i.e., at least one of the coefficients of this polynomial is non-zero. Thus $p_{B,i}(\varepsilon) \neq 0$ for all ε with $0 < \varepsilon < \delta_{B,i}$, where $\delta_{B,i}$ is the smallest positive root of $p_{B,i}$.

Taking the minimum of $\delta_{B,i}$ over all bases B (there are only finitely many) and for each basis over the indices i in the basis gives us δ as postulated in the first item.

$$\delta = \min\{\delta_{B,i}\ ;\ B \text{ is a basis and } i \in B\}\ .$$

We turn to the second item. Let δ be as determined in the first step and let B be an optimal basis of LP(ε) for some ε with $0 < \varepsilon < \delta$. Since an optimal basis is feasible, we have

$$x_i^* = \overline{b}_i + \sum_{1\le l\le m} \overline{a}_{il}\varepsilon^l \ge 0$$

for all $i \in B$. We even haven $x_i^* > 0$ since LP(ε) is non-degenerate. Since $\delta \ge \delta_{B,i} > \varepsilon$ there is no positive root of $p_{B,i}$ smaller than ε and hence $\overline{b}_i \ge 0$; observe that $\overline{b}_i = p_{B,i}(0) < 0$ and $x_i^* = p_{B,i}(\varepsilon) > 0$ would implies the existence of a positive root less than ε. Thus B is feasible for LP. Every non-basic variable has non-positive reduced cost and hence B is optimal for LP.

We turn to the third item. Consider an iteration of the simplex algorithm. Let j be an non-basic index with positive reduced cost. If $\overline{a}_{ij} \ge 0$ for all $i \in B$, the LP is unbounded. Otherwise, we look at all $k \in B$ with $\overline{a}_{kj} < 0$. We need to find the index which minimizes

$$\frac{p_{B,k}(\varepsilon)}{|\overline{a}_{kj}|}\ .$$

If there is only one index $k \in B$ with $\overline{a}_{kj} < 0$, the situation is clear.

Otherwise consider two such indices $i \in B$ and $k \in B$ with $\overline{a}_{ij} < 0$ and $\overline{a}_{kj} < 0$. We make the crucial observation that the polynomials $p_{B,i}(\varepsilon)/|\overline{a}_{ij}|$ and $p_{B,k}(\varepsilon)/|\overline{a}_{kj}|$ are distinct. Otherwise, the i-th and the k-th row of A_B^{-1} would be multiples of each other and hence the matrix A_B^{-1} would be singular. However, the matrix is non-singular. Let l be the smallest exponent for which the coefficients in the polynomials differ and assume that the coefficient is smaller in the first polynomial. Then the value of the first polynomial is smaller than the value of the second polynomial for all sufficiently small positive ε. This yields the following lexicographic rule for selecting the variable leaving the basis.

Assume $i \in B$ and $k \in B$ with $\overline{a}_{ij} < 0$ and $\overline{a}_{kj} < 0$. If $\overline{b}_i/|\overline{a}_{ij}| \neq \overline{b}_k/|\overline{a}_{kj}|$ it is clear which index is to preferred. If the values are equal we find the minimum l such that $\overline{a}_{il}/|\overline{a}_{ij}| \neq \overline{a}_{kl}/|\overline{a}_{kj}|$. i is to preferred if $\overline{a}_{il}/|\overline{a}_{ij}| < \overline{a}_{kl}/|\overline{a}_{kj}|$ and k is to be preferred otherwise. ∎

It is interesting to observe that the algorithm does not explicitly use ε. We only used it for the termination proof and in order to derive the lexicographic tie breaking rule.

1.2.3 Finding a First Feasible Basis: Phase I of the Simplex Method

Phase II of the simplex algorithm starts from an arbitrary feasible basis and terminates with an optimal basis. We still need to explain how to find a feasible basis.

We start with the system $Ax = b, x \in \mathbb{R}^n_{\geq 0}$. We multiply equations whose right hand side is negative by -1 and obtain a system whose right hand side is non-negative. So let us assume $b \geq 0$. We next add *artificial variables*, one for each constraint, and consider the problem

$$\min 1^T s \quad \text{subject to} \quad Ax + Is = b, \quad x \in \mathbb{R}^n_{\geq 0}, \quad s \in \mathbb{R}^m_{\geq 0} ,$$

where s is a vector of m non-negative artificial variables. We make the following observations.

- We may assume that the extended LP is non-degenerate. Otherwise, replace b by $b + \vec{\varepsilon}$.
- Taking s as the basic variables, x as the non-basic variables, we have a basic feasible solution $s = b + \vec{\varepsilon}$ and $x = 0$. Observe that the solution is feasible since $b \geq 0$ and $\varepsilon > 0$. We can therefore solve the problem by phase II of the simplex method.
- If the optimal solution has positive objective value our original problem has no solution, i.e., b is not in the positive span of the columns of A.
- So assume that the optimal solution has objective value zero. Thus $s = 0$ in the optimal solution and hence all artificial variables are non-basic in the optimal basis. This follows from our assumption that the extended LP is non-degenerate and hence basic variables are non-zero. We simple drop the artificial variables and obtain a basic feasible solution of the original problem.

Remark: If the extended LP is degenerate, the optimal basis may contain artificial variables. They can usually be swapped out of the optimal basis. We find it simpler to enforce non-degeneracy by perturbation.

1.3 Min Cost Network Flow

We apply linear programming and the simplex algorithms to a simple variant of the uncapacitated min cost flow problem in undirected networks. We will see that the problem is easily formulated as a linear program. We will also gain a better understanding of the simplex algorithm and we will introduce the concept of a dual problem.

We consider a simple version of the min cost flow problem in networks. We are given an undirected connected network $G = (V, E)$. Every node v has a demand/supply $b_v \in \mathbb{R}$. We call v a demand node if $b_v < 0$ and a supply node if $b_v > 0$. Supplies and demands balance, i.e., $\sum_v b_v = 0$. We want to ship goods from the supply nodes to the demand nodes. Every edge can carry an unlimited amount of goods and can be used in both directions. For an edge e we incur a cost of c_e for every unit shipped over the edge. In order to avoid trivialities, we assume $c_e > 0$ for every e.

It is convenient to turn the undirected network into a (bi)-directed network. For each edge of the undirected network we have two edges, one in each direction, in the directed network. The two directed edges have the same cost. From now on we use e to range over the edges of the directed network and we (re)-use E for the set of edges in the directed network.

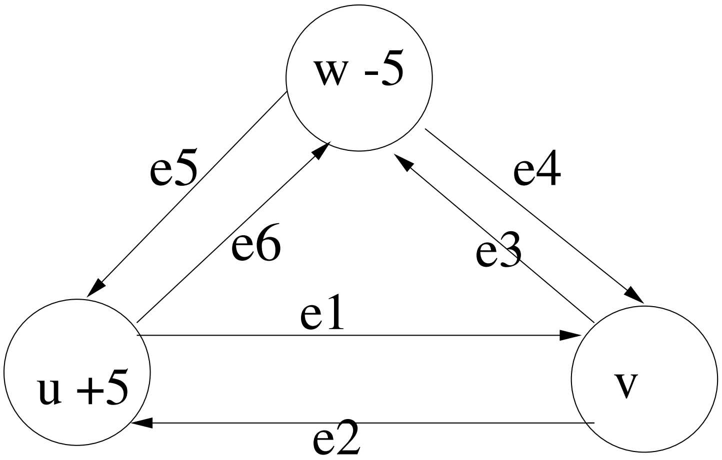

Figure 1.3: Edges e_1 and e_2 have cost a, e_3 and e_4 have cost b, and edges e_5 and e_6 have cost c. Node u supplies 5 units and node w demands 5 units. Here are some examples of feasible flows: (1) ship five units from u to v and then from v to w. The total cost is $5(a+b)$. (2) ship five units directly from u to w. The cost is $5c$. (3) ship six units from u to w and one unit from w to v and then on to u. The cost is $6c+a+b$.

For each edge we have a non-negative variable x_e to denote the amount of goods shipped over e. We call x_e the flow across e. The feasible flows are characterized by:

$$\text{for all } v \in V \qquad \sum_{e \in out(v)} x_e - \sum_{e \in in(v)} x_e = b(e) \ .$$

The goal is to minimize the total cost of the flow

$$\text{minimize} \quad \sum_e c_e x_e \ .$$

Figure 1.3 shows a small example. We have three nodes u, v, and w, and six edges e_1 to e_6; e_{2i+1} is the reversal of e_{2i} and both edges have the same cost.

$$\begin{array}{lrcl}
\text{minimize} & ax_1+ax_2+bx_3+bx_4+cx_5+cx_6 & & \\
\text{subject to} & x_1+x_6-x_2-x_5 & = & 5 \\
 & x_2+x_3-x_1-x_4 & = & 0 \\
 & x_4+x_5-x_3-x_6 & = & -5
\end{array}$$

One way of solving our example is to route five units each on e_1 and e_3. This corresponds to the following basic feasible solution:

$$\begin{array}{rlllll}
x_1 = 5 & +x_2 & & -x_5 & +x_6 \\
x_3 = 5 & & +x_4 & -x_5 & +x_6 \\
z = 5(a+b) & +2ax_2 & +2bx_4 & +(c-a-b)x_5 & +(c+a+b)x_6
\end{array}$$

Let us interpret what we see. We have five units of flow on e_1 and e_3. All other edges carry no flow. The cost of the flow in $5(a+b)$. What possibilities are there to modify the current solution. We could increase the flow on x_6. This forces us to increase the flow on x_1 and x_3.

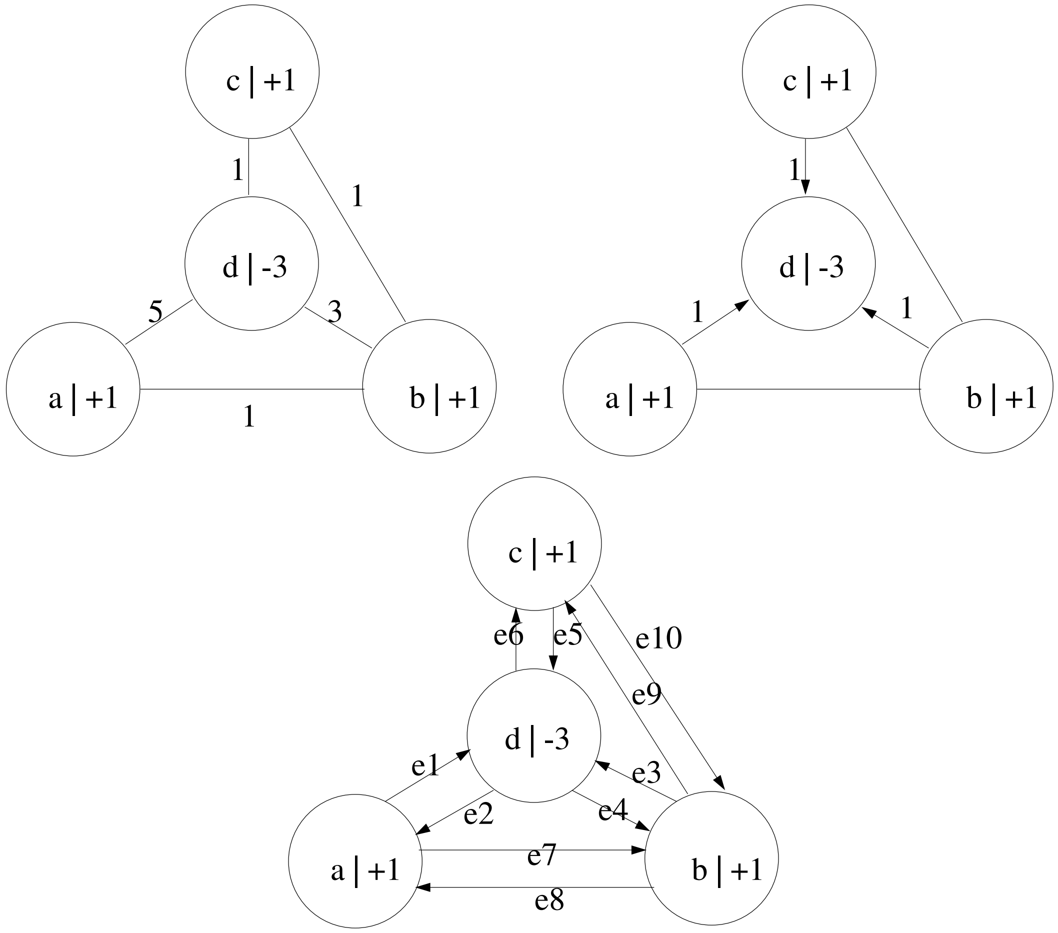

Figure 1.4: Nodes a, b, and c have surplus $+1$ and node d has demand -3. The cost of the edges is as shown, e.g., the uedge $\{ad\}$ has cost 5. The figure on the right indicates a solution. For each edge we indicate the value of the flow and the direction of the flow. For example, we are sending one unit from a to d. The cost of the flow shown is $5+3+1=9$. This is sub-optimal. The lower half shows the corresponding bidirected network.

The cost is $(c+a+b)$ per unit. This is counterproductive. We could increase the flow on e_4. An increase of the flow on e_4 would lead to an increase of the same amount on e_3. The cost is $2b$ for each unit sent. This is counterproductive. We could increase the flow on x_5. This would lead to a decrease on e_1 and e_3. The cost is $c-a-b$ per unit. It is therefore counterproductive to increase x_5 if $c > a+b$. The cost will decrease if $c < a+b$.

Assume $c < a+b$. Both constraints tell us to increase x_5 to 5.

Figure 1.4 shows a more complex example which I will do in class. I will not do the simple example in class.

- The LP formulation is as follows:

$$\text{minimize} \quad 5\cdot(x_1+x_2)+3\cdot(x_3+x_4)+1\cdot(x_5+\ldots+x_{10})$$

subject to

$$\begin{array}{lrcll}
\text{flow balance at } a: & x_1+x_7 & = & 1 & +x_2+x_8 \\
\text{flow balance at } b: & x_8+x_3+x_9 & = & 1 & +x_7+x_4+x_{10} \\
\text{flow balance at } c: & x_5+x_{10} & = & 1 & +x_6+x_9 \\
\text{flow balance at } d: & x_6+x_2+x_4 & = & -3 & +x_1+x_3+x_5
\end{array}$$

- We first observe that one of the equations is redundant, i.e., the rows of the constraint matrix are linearly dependent. This can be seen algebraically and semantically. Algebraically: summing the equations yields $x_1+\ldots+x_{10} = x_1+\ldots+x_{10}$ or $0=0$ and hence any equation is a linear combination of the others. Semantically: if the flow is balanced at all but one node then it is automatically balanced at all nodes. We drop the last equation and work with the first three equations.

- the variables x_1, x_7, x_3 do not form a basis. Indeed, none of them appears in the constraint for c and hence the equations cannot be solved for these variables.

- the variables x_1, x_4, x_5 form a basis, but this basis is not feasible for the given supplies/demands. Indeed the value for x_4 in the basic feasible solution is -1.

- the variables x_1, x_3 and x_5 form a feasible basis. Solving the first three equations for these variables yields:

$$\begin{array}{rcll}
x_1 & = & 1 & +x_2+x_8-x_7 \\
x_3 & = & 1 & +x_7+x_4+x_{10}-x_8-x_9 \\
x_5 & = & 1 & +x_6+x_9-x_{10} \\
z & = & 9 & +(5+5)x_2+(3+3)x_4+(1+1)x_6 \\
 & & & +(1-5+3)x_7+(1+5-3)x_8+(1-3+1)x_9+(1+3-1)x_{10} \\
 & = & 9 & +10x_2+6x_4+2x_6-1x_7+3x_8-1x_9+3x_{10}
\end{array}$$

  The solved form of the equations gives us a considerable amount of information.

  - First, we can read off a feasible solution. Send one unit each across the edges e_1, e_3 and e_5. The cost of this solution is 9.
  - Suppose we are using the edges e_1, e_3 and e_5 as the "back bone" of our flow and consider using one additional edge. The equations and the cost function tell us how we must change the flow and how the cost is going to be affected.
    * increasing the flow across e_2 leads to an increase by the same amount across e_1. The cost is 10 per unit of increase.
    * increasing flow along e_7 will decrease the flow along e_1 and increase the flow along e_3. The cost is $3+1-5$ per unit of increase. Observe that 4 is the cost of the path $e_7 \circ e_3$ and 5 is the cost of the path e_1
    * increasing the flow along e_9 will decrease the flow along x_3 and increase the flow along x_5. The cost is $1+1-3$ per unit increase. Observe that 2 is the cost of the $e_9 \circ e_5$ and 3 is the cost of the edge e_3.
    * the equations do not reveal yet that is worthwhile to use the path $e_7 \circ e_9 \circ e_5$ instead of the edge e_1.

- let us perform a basis update: remove either x_1 or x_3 from the basis and bring in either x_7 or x_9.

– if we remove x_1 from the basis and bring in x_7, we obtain (solve the first equation for x_7 and substitute the expression for x_7 into the equations for x_3, x_5 and z).

$$\begin{array}{rcll} x_7 & = & 1 & +x_2+x_8-x_1 \\ x_3 & = & 2 & +x_2+-x_1+x_4+x_{10}-x_9 \\ x_5 & = & 1 & +x_6+x_9-x_{10} \\ z & = & 8 & +x_1+(10-1)x_2+(6)x_4+(2)x_6 \\ & & & (3-1)x_8+(-1)x_9+(2)x_{10} \end{array}$$

Again the solved form of the equations gives us a lot of information. We are using the edges e_3, e_7, and e_5 as "back bone" edges. Increasing the cost along e_2 will increase the flow across e_7 and e_3 and hence has a cost of 9 per unit of increase. Increasing the flow across e_9 will decrease the flow across x_3 and increase it along x_5. The cost is $1+1-3$ per unit of increase. The maximal possible increase is 2. We make the increase by solving the second equation for x_9 and substituting the expression for x_9 into the other equations. We obtain:

$$\begin{array}{rcll} x_7 & = & 1 & +x_2+x_8-x_1 \\ x_9 & = & 2 & +x_2+-x_1+x_4+x_{10}-x_3 \\ x_5 & = & 3 & +x_6+x_2+-x_1+x_4-x_3 \\ z & = & 6 & +2x_1+8x_2+x_3+5x_4+2x_6+2x_8++1x_{10} \end{array}$$

Let us interpret the numbers. We have a flow of cost 6 and all non-basic variables have positive cost. The flow is optimal and it is the unique optimal flow. Increasing the flow across e_2 will lead to an increase along e_5, e_7 and e_9 and hence has a cost of 8 per unit of increase.

– if we remove x_3 from the basis and bring in x_9, we obtain (solve the second equation for x_9 and substitute the expression for x_9 into the equations for x_1, x_5 and z).

$$\begin{array}{rcll} x_1 & = & 1 & +x_2+x_8-x_7 \\ x_9 & = & 1 & +x_7+x_4+x_{10}-x_8-x_3 \\ x_5 & = & 2 & +x_6+x_7+x_4+-x_8-x_3 \\ z & = & 8 & +10x_2+x_3+5x_4+2x_6-2x_7+4x_8+2x_{10} \end{array}$$

We have a solution of cost 8. Increasing the flow across x_7 will decrease the flow across x_1 and increase the flow across x_9 and x_5. The cost is $1-5+1+1$ per unit of increase. Solving the first equation for x_7 and substituting the expression obtained into the other equations, we obtain

$$\begin{array}{rcll} x_7 & = & 1 & +x_2+x_8-x_1 \\ x_9 & = & 2 & +x_2+-x_1+x_4+x_{10}-x_3 \\ x_5 & = & 3 & +x_6+x_2+-x_1+x_4-x_3 \\ z & = & 6 & +2x_1+8x_2+x_3+5x_4+2x_6+2x_8++1x_{10} \end{array}$$

which is, of course, the same as above.

We next take an alternative view at the min cost flow problem. So far, we looked at the problem from the view of a trucking company (transport company, logistics company). Our goal was to design a flow of minimum cost that satisfies the supplies and demands.

We now consider the situation from the view of a trading company (merchant). What prices could we charge for our good at the charge locations of the network. What is a system of stable prices, i.e., a system of prices which gives no customer the incentive to move goods around. Let p_v be the price for one unit at node v. Clearly, if there is an edge $e = (u, v)$ then the price at v can be at most the price at u plus the cost of shipping our good from u to v, i.e.,

$$p_v \leq p_u + c(u, v) \quad \text{for all edges } e = (u, v) \ .$$

In our example, it would be unstable to have $p_d > p_a + 3$ since it costs only 3 Euro to ship a unit from a to b. In general, there are many systems of stable prices. In our example, the following systems of prices

$$p_a = p_b = p_c = p_d = 0 \qquad p_a = 0,\ p_b = 1,\ p_c = 2,\ p_d = 3$$

are stable.

Consider any system $(p_v)_{v \in V}$ of stable prices. In a system of stable prices, there is no incentive to ship goods around. A trading company could therefore operate as follows. It buys goods at supply nodes for a total cost of $\sum_{v \text{ is supply node}} p_v b_v$ and it sells goods at demand nodes for total revenues of $\sum_{v \text{ is demand node}} p_v |b_v| = -\sum_{v \text{ is demand node}} p_v b_v$; recall that demands are negative numbers. The total profit of our merchant is therefore

$$-\sum_{v \text{ is demand node}} p_v b_v - \sum_{v \text{ is supply node}} p_v b_v = \sum_v p_v(-b_v) \ .$$

In our example, the first system of stable prices gives a profit of 0 and the second system gives a profit of 6 Euros; the revenues are 9 Euros and the cost are 3 Euros. Of course, our trading company will attempt to maximize its profits. This leads to the problem

$$\text{maximize} \quad \sum_v p_v(-b_v) \quad \text{subject to } p_v \leq p_u + c(u, v) \text{ for all } e = (u, v) \ .$$

In our example, the shipping company can ship the goods for a total cost of 6 and the trading company can make a profit of 6 units. Is it a coincidence that we see twice the same number? **NO.**

Let $(x_e)_{e \in E}$ be a feasible flow and let $(p_v)_{v \in V}$ be a system of stable prices. For any path P, say from node u to node v we have $p_v \leq p_u + c(P)$. Any flow can be decomposed into a system of paths. This is most easily seen if the supplies and demands are integers. We can then view the flow as being sent around in units. Each unit traces a path from a supply node to a demand node. Let $P_1, \ldots P_k$ be a decomposition of a flow into paths. Since each path carries a unit, we have

$$\text{cost of the flow} = \sum_i c(P_i) \ .$$

Since we are having a system of fair prices, we have

$$c(P_i) \geq p(v_i) - p(u_i) \ ,$$

where P_i is a path from u_i to v_i. Thus

$$\text{cost of flow} = \sum_i c(P_i) \geq \sum_i (p(v_i) - p(u_i)) \ .$$

A node v with surplus is the source of b_v paths and a node with demand b_v is the destination of $-b_v$ paths (recall that demands are negative). Thus $\sum_i (p(v_i) - p(u_i)) = \sum_v (-b_v) p_v$ and hence

$$\text{cost of flow} \geq \sum_v (-b_v) p_v \ .$$

We conclude that any system of stable prices provides us with a lower bound on the cost of our transport problem.

We next give a "physical" proof. We construct a physical model of our network by assigning a height to each node. We put a node v at height p_v and think of edges as pipes transporting water. Viewed from the outside we have a system into which water enters at the supply nodes and from which water is removed at the demand nodes. Viewed from the inside the water moves through the pipes.

Since $p_v \leq p_u + c(u,v)$ the height difference between the target and the source node of any edge is bounded by the cost of the edge. In other words, if we move one unit from u to v the gain in "potential energy" is at most the cost of the edge. Thus a feasible flow $(x_e)_{e \in E}$ can increase the potential energy of the system by at most the cost of the flow. On the other hand, we have b_v units entering the system at supply node v (for a total potential energy of $\sum_{v \text{ is supply node}} p_v b_v$) and we have $-b_v$ units leaving the system at demand node v (for a total potential energy of $-\sum_{v \text{ is demand node}} p_v b_v$). Thus the gain in potential energy is $\sum_v (-b_v) p_v$ and hence

$$\sum_v (-b_v) p_v = \text{gain in potential energy} \leq \text{cost of flow} = \sum_e c_e x_e \ .$$

We show next that the optimal solution to the transport problem and to the trading problem have the same optimal solution. Assume that $(x_e)_{e \in E}$ is an optimal flow. We may also assume that the flow carrying edges form a tree T in the underlying undirected graph. We simple use the edges of non-zero flow to fix the prices, i.e, if $x_e > 0$ and $e = (u,v)$, we set $p_v = p_u + c(u,v)$. This is a simple computation. We simply fix the price for an arbitrary node and propagate the prices along the edges of T. In our example, we might fix the price at node c to zero. Propagation yields $p_d = 1$, $p_b = -1$ and $p_a = -2$. We may again visualize the situation physically. We put node v at height p_v. Then every flow carrying edge (u,v) goes upwards and covers a vertical distance of exactly $p_v - p_u$. Optimality means that it does not pay off to activate any of the edges that carry no flow. Let (a,b) be any such edge. Then we must have $p_b \geq p_u + c(a,b)$ because otherwise, it would be beneficial to use the edge $e = (a,b)$. We conclude that the system of prices is stable.

We next look at the problem in general. The material till the end of the section may be skipped and there is no particular organization to it. But, there is also no harm to carry on.

1. In an optimal solution at most one of e and e^R carries flow; here e^R denotes the reversal of e. Indeed, if both carry flow let $f = \min(x_e, x_{e^R}$. We reduce the flow across e and e^R by f and save $f x_e$ Euro.

2. More generally, if C is any directed cycle, at least one edge in C does not carry flow. Otherwise, let f be the minimum amount carried by an edge in C. We reduce all flows by f.

3. For a directed path p let $c(p) = \sum_{e \in p} c_e$ be the cost of p. Consider two path p and q sharing source and target. If $c(p) < c(q)$, at least one edge of q carries no flow. Otherwise. let f be the minimum flow carried by an edge in q. We reduce the flow along q by f and increase the flow along p by f.

4. Let C be a directed cycle and assume that for each edge of C either the edge or its reversal carries flow. Let C^+ be the set of edges of C that carry flow and let C^- be the set of edges in C whose reversal carries flow. Let c^+, respectively c^-, be the total cost of the edges in C^+, C^- respectively. Assume $c^+ \le c^-$, the other case is similar, and let f be the minimum flow across any edge in C^+ or C^-. We increase the flow on all edges in C^+ by f and decrease the flow on all edges in C^-. This preserves feasibility and saves $f(c^- - c^+)$ Euros. Draw a figure.

5. The previous item implies that there is an optimal solutions in which in every directed cycle there is at least one edge such that neither the edge nor its reversal carries flow. This implies that the set of edges carrying flow forms a tree in the underlying undirected graph.

Let us now turn to the simplex algorithm. We start with a closer look at the constraint matrix A. We have n constraints, one for each node of G and $2m$ variables, two for each edge of the undirected graph. The constraint matrix has two non-zero entries per column. For a directed edge $e = (u, v)$, the entry is $+1$ in row u and -1 in row v. The rows of the constraint matrix are not independent. Indeed, summing all rows gives us the zero-vector. We may therefore delete an arbitrary constraint.

We leave it to the reader to prove that the constraint matrix has rank $n-1$. This uses the fact that the network is connected.

We reuse A for the constraint matrix with one row deleted. What are the non-singular submatrices of A?

Lemma 1 *A basis B is a set of $n-1$ directed edges forming a tree in the underlying undirected graph.*

Proof: We first show that any such set of edges allows us to define flows. We simply work our way in from the leaves of the tree. The flow obtained in this way is not necessarily feasible, as we may have the wrong direction of an edge.

Consider now a set of $n-1$ edges that does not form a tree. Then we can partition V into two sets V_1 and V_2 such that no edge connecting the two sets is in the selected set of edges. Next selects demands/supplies such that there most be flow between the sets. However, we cannot solve the balance equations for the selected variables. Thus A_B is not invertible. ∎

In the dual problem, we have a variable y_v for each vertex. The variable is unconstrained since the constraints in the primal are equality constraints. We have $2m$ constraints, one for each directed edge. The constraint for the directed edge $e = (u, v)$ is $y_u + c_e \ge y_v$.

1.4 Linear Programming Duality

Duality is a particularly beautiful and useful part of linear programming theory. For every linear program of maximization type there is a linear program of minimization type with the same objective value. The optimal solutions for the two programs are closely related. Duality theory gives rise to combinatorial algorithms for some linear programming problems.

1.4.1 Algebraic Motivation of Duality

Consider

$$\text{maximize} \quad 5x_1 + 7x_2$$

subject to

$$x_1 + 4x_2 \le 4 \quad x_1 + x_2 \le 2 \quad x_1, x_2 \in \mathbb{R}_{\ge 0} .$$

There is a simple way for deriving an upper bound on the objective value. Multiplying the first equation by four and the second equation by one and adding yields

$$5x_1 + 17x_2 \le 18$$

and hence the objective value of the LP is bounded by 18. We could also multiply the first equation by 2 and the second by 3 instead. This would give

$$5x_1 + 11x_2 \le 14 .$$

More generally, any pair of non-negative numbers y_1 and y_2 with $y_1 + y_2 \ge 5$ and $4y_1 + y_2 \ge 7$ gives us an upper bound. The bound is $4y_1 + 2y_2$. We best bound that we can obtain in this way is

$$\min \{4y_1 + 2y_2 \ ; y_1 + y_2 \ge 5,\ 4y_1 + y_2 \ge 7\ y_1, y_2 \in \mathbb{R}_{\ge 0}\} .$$

The optimal value of the primal LP is $34/3$ (at $x^* = (4/3, 2/3)$). The optimal value of the dual LP is $34/3$ (at $y^* = (2/3, 13/3)$). We generalize.

Consider an LP of the form

$$\text{(PLP)} \qquad \text{maximize} \quad c^T x \quad \text{subject to} \quad Ax \le b, \quad x \in \mathbb{R}^n_{\ge 0} .$$

We tagged the equation with PLP (primal linear program) for further reference. There is a simple way for deriving upper bounds for the optimal objective value of PLP by taking non-negative linear combinations of the inequalities dominating the cost function. More precisely, we consider $y^T = (y_1, \ldots, y_m) \ge 0$ with $y^T A \ge c^T$. Then

$$c^T x \le y^T A x \le y^T b ,$$

where the first inequality uses $x \ge 0$ and the second inequality uses $y \ge 0$. Any $y \in \mathbb{R}^m_{\ge 0}$ with $c^T \le y^T A$ gives us an upper bound on the objective value of PLP, namely $y^T b$. Finding the best such bound is just another linear program.

$$\text{(DLP)} \qquad \text{minimize} \quad y^T b \quad \text{subject to} \quad y^T A \ge c^T, \quad y \in \mathbb{R}^m_{\ge 0} .$$

DLP is called the linear program dual to PLP.

Definition 1 *Given a linear program*

(PLP) $\qquad$ *maximize* $\quad c^T x \quad$ *subject to* $\quad Ax \leq b, \quad x \in \mathbb{R}^n_{\geq 0}$.

its dual is given by

(DLP) $\qquad$ *minimize* $\quad y^T b \quad$ *subject to* $\quad y^T A \geq c^T, \quad y \in \mathbb{R}^m_{\geq 0}$.

The next theorem states what we already know. That the dual gives upper bounds for the primal and the primal gives lower bounds for the dual. This is called the weak duality theorem of linear programming.

Theorem 8 (Weak Duality Theorem) *If $x^* \in \mathbb{R}^n_{\geq 0}$ is a feasible solution of PLP and $y^* \in \mathbb{R}^m_{\geq 0}$ is a feasible solution of DLP then*

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

Proof: We have

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

where the first inequality follows from $c^T \leq {y^*}^T A$ and $x^* \geq 0$ and the second inequality follows from $Ax^* \leq b$ and $y^* \geq 0$. ∎

We even have equality as we will see in Section 1.4.3.

1.4.2 Rules for Forming the Dual

In our theoretical treatment of duality we will always consider LPs in inequality form with non-negative variables. In the applications, we will use equalities, inequalities, unconstrained and constrained variables. Of course, duality applies to any form of LP. In this section we state general rules for forming duals. Consider a primal problem

- maximize $c^T x$ with
- n variables $x_1, \ldots, x_n$ and
- m constraints $a_i^T \heartsuit_i b_i$ with $\heartsuit_i \in \{\leq, =, \geq\}$ for $1 \leq i \leq m$.
- a primal variable x_i is either unconstrained or non-negative ($x_i \geq 0$ or non-positive ($x_i \leq 0$).

The dual has

- m variables $y_1, \ldots, y_m$ corresponding to the m constraints of the primal.
- y_i is unconstrained if the i-th constrained of the primal is an equality constraint and it is constrained otherwise. If $\heartsuit_i = \leq$ then $y_i \geq 0$ and if $\heartsuit_i = \geq$ then $y_i \leq 0$. Observe that the senses are reversed.
- We have n constraints one for each variable of the primal. The i-th dual constraint is $y^T A_i \diamondsuit_i c_i$ where A_i is the i-th column of the constraint matrix. The constraint is an equality constraint if x_i is unconstrained. Otherwise, we have $\diamondsuit_i = \geq$ if $x_i \geq 0$ and $\diamondsuit_i = \leq$ if $x_i \leq 0$. Observe that the sense is not reversed.

- the objective function is $y^T b$ and we are minimizing.

We prove weak duality. Let x^* and y^* be solutions of the primal and dual, respectively. Then

$$c^T x^* \leq y^T A x^* \leq y^T b \; .$$

We first argue the second inequality.

- If $a_i x^* = b_i$ then $y_i a_i x^* \leq y_i b_i$ for an arbitrary y_i.
- If $a_i x^* \leq b_i$ then $y_i a_i x^* \leq y_i b_i$ for an arbitrary non-negative y_i.
- If $a_i x^* \geq b_i$ then $y_i a_i x^* \leq y_i b_i$ for an arbitrary non-positive y_i.

The second inequality follows by summing over i. We next argue the first inequality.

- If $y^T A_i = c_i$ then $c_i x_i \leq y^T A_i x_i \leq c_i x_i$ for an arbitrary x_i.
- If $y^T A_i \leq c_i$ then $c_i x_i \leq y^T A_i x_i \leq c_i x_i$ for an arbitrary non-positive x_i.
- If $y^T A_i \geq c_i$ then $c_i x_i \leq y^T A_i x_i \leq c_i x_i$ for an arbitrary non-negative x_i.

The equality follows by summation over i.

Let us give an example. In the introduction we formulated the maximum weight bipartite matching problem as a linear program (we assume for the moment that we know already that the LP formulation guarantees integrality of the primal variables.). We have $|E|$ variables, one for each edge of $G = (A \dot{\cup} B, E)$. We use x_e to denote the variable for edge e and use c_e for the weight of edge e. For a node v we use $\delta(v)$ to denote the set of edges incident to v.

$$\begin{array}{llcll}
\text{maximize} & \sum_e w_e x_e & & & \\
\text{subject to} & \sum_{e \in \delta(v)} x_e & \leq & 1 & \text{for all } v \in V \\
 & x_e & \geq & 0 & \text{for all } e \in E
\end{array}$$

In the introduction we also had the constraints $x_e \leq 1$. These constraints are redundant and hence I dropped them.

In the primal we have one constraint for each vertex of G and hence it the dual we have one variable for each node of G. We use y_v to denote the variable corresponding to node v. The sense of the constraints in the primal is $\leq$ and hence the variables in the dual are non-negative ($y_v \geq 0$).

The variables of the primal correspond to edges of G and hence we have one constraint for each edge in the dual. Our variables in the primal are non-negative and hence all constraints in the dual will be of the type $\geq$. Our constraint matrix A is a zero-one matrix. It has a 1 in entry (v, e) (= the entry corresponding to vertex v and edge e iff e is incident to v). Thus the dual constraint corresponding to edge e involves exactly two variables, namely the variables y_a and y_b corresponding to the endpoints a and b of e. The constraint is $y_a + y_b \geq c_e$. We obtain

$$\begin{array}{llcll}
\text{minimize} & \sum_v y_v & & & \\
\text{subject to} & y_a + y_b & \geq & w_e & \text{for all } e = ab \in E \\
 & y_v & \geq & 0 & \text{for all } v \in V
\end{array}$$

and the miracle from the introduction is not a miracle anymore but a theorem.

1.4.3 Strong Duality Theorem and Complementary Slackness

We prove important properties of duality. The first theorem tells us that primal and dual play the same role. It really does not matter which of the two problems is called the dual and which one the primal.

Theorem 9 *The dual of the dual is the primal*

Proof: The proof is by rewriting the LP a few times. The dual is given by

$$\text{minimize} \quad y^T b \quad \text{subject to} \quad y^T A \geq c^T, \quad y \in \mathbb{R}^m_{\geq 0} .$$

We turn it into a maximization problem by multiplying the cost vector b by -1 and we change the sense of the inequalities by multiplying A and c by -1. We obtain:

$$\text{maximize} \quad y^T(-b) \quad \text{subject to} \quad y^T(-A) \leq (-c)^T, \quad y \in \mathbb{R}^m_{\geq 0} .$$

Transposing matrix and right-hand side gives us further

$$\text{maximize} \quad (-b)^T y \quad \text{subject to} \quad (-A)^T y \leq (-c), \quad y \in \mathbb{R}^m_{\geq 0} .$$

We can now go to the dual and obtain:

$$\text{minimize} \quad x^T(-c) \quad \text{subject to} \quad x^T(-A)^T \geq (-b)^T, \quad y \in \mathbb{R}^m_{\geq 0} .$$

Multiplying everything by -1 changes the sense of the optimization and the sense of the inequalities and transposing the constraints moves x to the right side where we want it to be. We obtain

$$\text{maximize} \quad c^T x \quad \text{subject to} \quad Axy \leq b, \quad y \in \mathbb{R}^m_{\geq 0} .$$

∎

We next come to the strong duality theorem of linear programming.

Theorem 10 (Strong Duality) *If PLP is feasible and has finite optimum, DLP is feasible and has finite optimum. Furthermore, if $x^* \in \mathbb{R}^n_{\geq 0}$ is an optimal solution of PLP and $y^* \in \mathbb{R}^m_{\geq 0}$ is an optimal solution of DLP then*

$$c^T x^* = {y^*}^T b .$$

Proof: We start by converting (PLP) into equality form by adding a slack variable for each constraint.

$$\text{(PLP1)} \qquad \text{maximize} \quad c^T x \quad \text{subject to} \quad Ax + Is = b, \quad x \in \mathbb{R}^n_{\geq 0},\ s \in \mathbb{R}^m_{\geq 0} .$$

PLP1 has a finite optimum iff PLP has. Consider an optimal basis B of PLP1. It corresponds to a a subset of m variables such that the submatrix of (A, I) selected by them is invertible. The submatrix consists of say n_1 columns of A and $m - n_1$ columns of I. Let x_B be the n_1 basic variables among the x and s_B be the basic variables among the s. The basic slack variables select a subset $m - n_1$ of the rows. So let us split the rows accordingly. We obtain

$$\begin{array}{cccccc} & n_1 & n-n_1 & n_1 & m-n_1 & \\ n_1 & A_{11} & A_{12} & I & O & b_1 \\ m-n_1 & A_{21} & A_{22} & O & I & b_2 \\ & c_1 & c_2 & & & \end{array}$$

or

$$\begin{array}{rcl} A_{11}x_B + A_{12}x_N + Is_N + Os_B & = & b_1 \\ A_{21}x_B + A_{22}x_N + Os_N + Is_B & = & b_2 \end{array}$$

In the basic solution we have $x_B^* = A_{11}^{-1}b_1$ and $s_B^* = b_2 - A_{21}x_B^*$.

Remark: The remark is not required for the proof. However, it tells us about the feasibility criterion of a basis of an LP in inequality form. In a basic solution for an LP in inequality form we select a subset of the constraints and turn them into equalities. We then solve for a subset of the variables of appropriate cardinality.

The basis is feasible if $x_B^* \geq 0$, i.e., $A_{11}^{-1}b_1 \geq 0$ and $s_B^* \geq 0$, i.e., $A_{21}x_B^* \leq b_2$ or equivalently $A_{21}A_{11}^{-1}b_1 \leq b_2$. In other words, the basis is feasible if the basic variables are non-negative and the remaining inequalities are satisfied. End of remark.

The objective value is $z = c_1^T x_B^* = c_1^T A_{11}^{-1} b_1$. The dependency of the objective value on the non-basic variables is given by

$$\begin{array}{rcl} z & = & c_1^T A_{11}^{-1}(b_1 - A_{12}x_N - Is_N) + c_2^T x_N \\ & = & c_1^T A_{11}^{-1} b_1 + (c_2^T - c_1^T A_{11}^{-1} A_{12})x_N - c_1^T A_{11}^{-1} s_N \end{array}$$

Since our basis is optimal we have

$$(*)\ c_2^T - c_1^T A_{11}^{-1} A_{12} \leq 0 \quad \text{and} \quad (**)\ c_1^T A_{11}^{-1} \geq 0\ .$$

We next consider the dual problem. Let y_B be the variables corresponding to the constraints which are satisfied with equality in the primal, i.e., the dual reads

$$\text{minimize} \quad y_B^T b_1 + y_N^T b_2 \quad \text{subject to} \quad y_B^T A_{11} + y_N^T A_{21} \geq c_1 \quad y_B^T A_{12} + y_N^T A_{22} \geq c_2 \quad y \in \mathbb{R}^m_{\geq 0}\ .$$

We consider the solution obtained by setting $y_N^* = 0$, turning the first set of inequalities in equalities and solving for y_B. We obtain $(y_B^*)^T = c_1^T A_{11}^{-1}$. Then $(y_B^*)^T \geq 0$ by (**).

The second set of inequalities is satisfied since

$$(y_B^*)^T A_{12} + (y_N^*)^T A_{22} = (y_B^*)^T A_{12} = c_1^T A_{11}^{-1} A_{12} \geq c_2^T$$

by (*). Thus we have a feasible solution of the dual.

We finally observe that

$$(y_B^*)^T b_1 + (y_N^*)^T b_2 = (y_B^*)^T b_1 = c_1^T A_{11}^{-1} b_1 = c_1^T x_B^* = c_1^T x_B^* + c_2^T x_N^*$$

∎

As an almost immediate consequence we get the following result about feasibility of primal and dual.

Theorem 11 *The following four cases are possible and exhaustive :*

- *PLP and DLP are both feasible and have a finite optimum.*
- *PLP is unbounded and DLP is infeasible.*
- *PLP is infeasible and DLP is unbounded.*
- *PLP and DLP are both infeasible.*

Proof: If one of the problems has a finite optimum, the other problem has by strong duality. So either both problems have a finite optimum or none has. If a problem does not have a finite optimum it is either infeasible or unbounded. Weak duality excludes the case that both problem are unbounded. We have now shown that the four cases listed exhaust the possibilities.

It remains to show that all four cases are indeed possible. We give examples.

PLP and DLP have finite optima:

$$\max\{x_1\ ; x_1 \le 1,\ x_1 \in \mathbb{R}_{\ge 0}\} = \min\{y_1\ ; y_1 \ge 1,\ y_1 \in \mathbb{R}_{\ge 0}\}$$

PLP is unbounded and DLP is infeasible:

$$\max\{x_1\ ; x_2 \le 1,\ x_1, x_2 \in \mathbb{R}_{\ge 0}\} \quad \text{and} \quad \min\{y_1\ ; 0 \cdot y_1 \ge 1,\ 1 \cdot y_1 \ge 0,\ y_1 \in \mathbb{R}_{\ge 0}\}$$

PLP is infeasible and DLP is unbounded: symmetric to the preceding case.

PLP and DLP are both infeasible:

$$\max\{1 \cdot x_1\ ; 0 \cdot x_1 \le -1,\ x_1 \in \mathbb{R}_{\ge 0}\} \quad \text{and} \quad \min\{-1 \cdot y_1\ ; 0 \cdot y_1 \ge 1\ y_1 \in \mathbb{R}_{\ge 0}\}$$

∎

Another important property of optimal primal-dual pairs is complementary slackness. Let $s = b - Ax$ be the vector of slack variables of the primal and let $t^T = y^T A - c^T$ be the surplus variables of the dual.

Theorem 12 (Complementary Slackness) *let x^* and y^* be feasible solutions of PLP and DLP, respectively: x^* and y^* are optimal solutions iff $x_j^* t_j^* = 0$ for all j and $y_i^* s_i^* = 0$ for all i. In words, if a primal variable is non-zero, the corresponding inequality is tight in the dual and if a dual variable is non-zero the corresponding constraint in the primal is tight.*

Proof: Using the definitions of s and t, we have

$$\begin{aligned} c^T x^* &= ((y^*)^T A - (t^*)^T) x^* \\ &= (y^*)^T A x^* - (t^*)^T x^* \\ &= (y^*)^T (b - s^*) - (t^*)^T x^* \\ &= (y^*)^T b - (y^*)^T s^* - (t^*)^T x^* \ . \end{aligned}$$

Thus if $(y^*)^T s^* - (t^*)^T x^* = 0$ then $c^T x^* = (y^*)^T b$ and hence x^* and y^* are optimal by weak duality. Conversely, if x^* and y^* are optimal, they yield the same objective value by strong duality and hence $(y^*)^T s^* + (t^*)^T x^* = 0$. The claim follows since all our variables (primal, dual, slack, surplus) are non-negative. ∎

The next two lemmas apply duality to the problem of proving that a system of equations does not have a solution.

show a simple example in two variables. $x+y=2$ and $2x+2y=3$. Multiplying the first equality by 2 and subtracting the result from the second equation yields $0=-1$ and hence the system is infeasible. Such a proof exists always.

Lemma 2 *If $Ax=b$ is infeasible iff there is a y such that $y^TA=0$ and $y^Tb\neq 0$.*

Proof: Assume $Ax^*=b$ and let y be any vector with $y^TA=0$. Then $y^Tb=y^TAx^*=0x^*=0$. This proves the direction from right to left.

So assume that $Ax=b$ is infeasible. Then $\max\left\{0\cdot x\,;Ax=b\right\}$ is infeasible and hence its dual $\min\left\{y^Tb\,;y^TA=0\right\}$ is either unbounded or infeasible. It is feasible since the $y=0$ is a feasible solution. Thus it is unbounded and the desired y exists. ∎

There is a similar statement for inequalities. $x+y\leq 2$ and $-x-y\leq -3$. Adding the inequalities yields $0\leq -1$ and hence the system is infeasible.

Lemma 3 (Farkas Lemma) *Assume that $Ax\leq b$ is infeasible*

- *$Ax\leq b$ is infeasible iff there is $y\geq 0$ with $y^TA=0$ and $y^Tb<0$.*
- *If $Ax\leq b$ is infeasible then there is a subset of $n+1$ inequalities which are infeasible.*

Proof: Assume $Ax^*\leq b$ and let $y\geq 0$ be any vector with $y^TA=0$. Then $y^Tb\geq y^TAx^*=0x^*=0$. This proves the direction from right to left.

So assume that $Ax\leq b$ is infeasible. Then $\max\left\{0\cdot x\,;Ax\leq b\right\}$ is infeasible and hence its dual $\min\left\{y^Tb\,;y^TA=0,y\geq 0\right\}$ is either unbounded or infeasible. It is feasible since the $y=0$ is a feasible solution. Thus it is unbounded and the desired y exists. This proves the direction from left to right.

The simplex algorithm tells us that the dual solution has an unbounded ray y^*+td. In this ray at most $n+1$ components are non-zero. ∎

Show example for the second part. For $n=1$, each inequality is of the form $x\leq b$ or $x\geq b$ for some constant b. Clearly, there is a strongest upper bound and a strongest lower bound. The system is unsolvable iff they contradict each other.

Already for two variables, the claim is surprising, in particular, in its contrapositive form. Consider a system of inequalities in n variables. If any subsystem of $n+1$ constraints is feasible, the entire system is.

1.4.4 Geometric Interpretation of Duality I

Consider the problem

$$\text{(LP)} \qquad \max\left\{c^Tx\,;Ax\leq b\right\}\ .$$

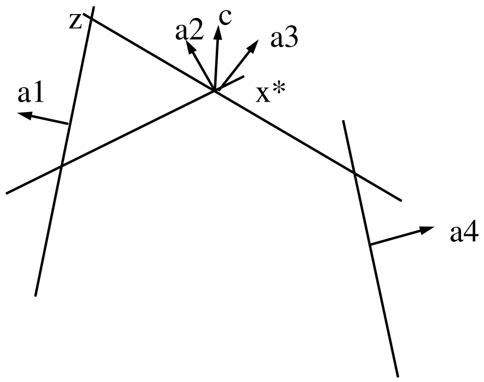

Figure 1.5: The optimization direction c is a non-negative linear combination of the a_2 and a_3 and the optimal point x^* lies on the constraints $a_2^T x = b_2$ and $a_3^T x = b_3$. We could also represent c as a non-negative combination of a_1 and a_3. This would give us the objective value at z as an upper bound. This bound is not the best possible, since x^* lies strictly below the hyperplane $a_1^T x = b_1$.

The feasible set $P = \{x\,;\,Ax \le b\}$ is a polyhedron. Finding the maximum of (LP) is tantamount to finding the hyperplane orthogonal to c (i.e., with equation $c^T x = \beta$) tangent to P, and maximal in direction c (with the maximal value of β).

Suppose that the maximum is finite, say with value β^* and let $x^* \in P$ be an optimal point in P. The optimal point will satisfy some of the inequalities defining P with equality, say $a_1^T x \le b_1$, $\ldots$, $a_k^T x \le b_k$ are the inequalities which x^* satisfies with equality.

Geometric intuition (see Figure 1.5) tells us that the vector c is a *non-negative* linear combination of the vectors a_1 to a_k, i.e., $c = \sum_i \lambda_i a_i$ with $\lambda_i \ge 0$ for all i. Then $\beta^* = c^T x^* = \sum_i \lambda_i a_i^T x^* = \sum_i \lambda_i b_i$. Thus

$$\max\left\{c^T x\,;\,Ax \le b\right\} = \beta^* = \sum_i \lambda_i b_i \ge \min\left\{y^T b\,;\,y \ge 0 \text{ and } y^T A = c\right\}\ ,$$

where the inequality follows from the fact that λ_i are a feasible solution of the minimization problem. Thus

$$\max\left\{c^T x\,;\,Ax \le b\right\} = \min\left\{y^T b\,;\,y \ge 0 \text{ and } y^T A = c\right\}$$

as the we know already from weak duality that we have $\le$.

We can also *see* complementary slackness. If $a^T x^* < b$ for some inequality in $Ax < b$, the corresponding λ is set to zero in the construction above.

1.4.5 Geometric Interpretation of Duality II (this may be skipped)

Give some more examples in the plane.

The remainder of this section assumes $n = 2$. We assume for simplicity that $c^T = (0,1)$, i.e., we look for a feasible point (x,y) with maximal y-coordinate. We also assume that the feasible region is bounded and that there is no horizontal constraint, i.e., the coefficient of x is non-zero in every constraint. In this situation the optimal solution is unique and is given as the intersection of the bounding lines of two of our half-spaces. The optimal solution is the *highest feasible point*.

We next introduce a dual concept. Two of our half-spaces are said to form a *V-combination* if their intersection is bounded from above. The intersection of two half-spaces is either a half-space or a wedge (Keil).

In the former case the boundaries of the half-spaces are parallel and since we have no horizontal constraints the intersection is unbounded in the positive y-direction. A wedge is bounded by two rays emanating from a common point, the tip of the wedge. The height of a wedge is the height (y-coordinate) of the tip. In a V-combination the boundary rays point downwards and the tip is the highest point in the wedge. We are interested in the V-combination of minimal height. We observe

- the height of any V-combination is bounded from below by the height of any feasible point (weak duality). This is obvious since the height of the V-combination is simply the value of our LP where we drop all constraints but the ones forming the V-combination.

- the height of the lowest V-combination is equal to the height of the highest feasible point (strong duality). This is obvious if we consider the V-combination consisting of the two half-spaces defining the optimal feasible point.

Let us try to express the problem of finding the minimal V-combination as a linear program. Two halfspaces $a_i x \le b_i$ and $a_j x \le b_j$ form a V-combination iff their normal vectors a_i and a_j point upwards and one points to the left and one points to the right. In other words, iff we can write c (recall $c^T = (0,1)$) as a positive linear combination of a_i and a_j, i.e.,

$$c = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = a_i y_i + a_j y_j \quad \text{and} \quad y_i \ge 0, y_j \ge 0$$

or in matrix notation

$$c^T = (0,1) = (y_i, y_j) \begin{pmatrix} a_i^T \\ a_j^T \end{pmatrix} \quad \text{and} \quad y_i \ge 0, y_j \ge 0 \; .$$

We have written the equation as row vectors in order to emphasize that we are combining rows of the constraint matrix in order to form the cost vector. Let B be the matrix formed by row vectors a_i^T and a_j^T. Then $y'^T = (y_i^*, y_j^*) = c^T B^{-1}$ is the unique solution to this system.

What is the height of the V-combination of half-spaces i and j? It is

$$y_i^* b_i + y_j^* b_j$$

as we show next. Let $b' = \begin{pmatrix} b_i \\ b_j \end{pmatrix}$ be the vector formed by the righthand sides of the half-spaces in the V-combination. The tip of the wedge is $x^* = (x_1, x_2) = B^{-1} b'$ and its height is $c^T x^* = c^T B^{-1} b' = y'^T b'$.

This suggests to formulate the problem of finding a minimal V-combination as follows.

$$\text{minimize} \quad y^T b \quad \text{subject to} \quad y^T A = c \quad y \in \mathbb{R}^m_{\ge 0} \; .$$

Actually, I have cheated slightly. A V-combination is a combination of two half-spaces and now I allow combinations of more than two half-spaces. Thus the minimum could become smaller. We will see later that it does not. Can you see it now?

1.5 The Dual Simplex Method

The idea is simple to explain.

Solve an LP by running the primal simplex algorithm on the dual LP. Do so without explicitly mentioning the dual.

We consider an LP with non-negative variables and constraints in equality form.

(LP) $$\text{maximize} \quad c^T x \quad \text{subject to} \quad Ax = b, \quad x \in \mathbb{R}^n_{\geq 0}\ .$$

As in the primal simplex method we partition the variables into basic and non-basic, solve the constraint system for the basic variables and express the objective function as a linear function of the non-basic variables.

$$\begin{aligned} x_i &= \overline{b}_i + \sum_{j \in N} \overline{a}_{ij} x_j \qquad \text{for } i \in B \\ z &= \overline{z} + \sum_{j \in N} \overline{c}_j x_j \end{aligned}$$

In the primal simplex algorithm we insisted that the basis is feasible, i.e., $\overline{b} \geq 0$. We iterated until all reduced costs were non-positive.

In the dual simplex algorithm we maintain that the reduced costs are non-positive (the basis is said to be dual feasible) and iterate until the basis is also feasible. So our invariant is

$$\overline{c}_N \leq 0\ .$$

Here is what we do in an iteration (the reader should go back to the formulation of phase II of the primal simplex method to see the similarity (duality)).

- If $\overline{b}_i \geq 0$ for all $i \in B$, the current basis is optimal. $x_i^* = \overline{b}_i$ for $i \in B$ and $x_j^* = 0$ for $j \in N$ is an optimal solution and $\overline{z}$ is the optimal objective value.
- Otherwise, let $i \in B$ be arbitrary with $\overline{b}_i < 0$. In the primal simplex method, we select a non-basic variable with negative reduced cost. It will enter the basis. Now we select a basic variable with negative value. It will leave the basis.
  - If $\overline{a}_{ij} \leq 0$ for all $j \in N$, the LP is infeasible since the (non-basic) variables are constrained to be non-negative and the equation for x_i shows that there is no way to give x_i and all non-basic variables non-negative values.
  - Otherwise, we need to choose one of the $j \in N$ with $\overline{a}_{ij} > 0$ to enter the basis. Which one? Assume for the moment, that we move to the basis $B' = B \setminus i \cup j$. Then we obtain the dictionary (this is exactly as in the primal simplex method):

$$\begin{aligned} x_j &= \frac{-\overline{b}_i}{\overline{a}_{ij}} \qquad\qquad + \frac{1}{\overline{a}_{ij}} x_i + \sum_{l \in N \setminus j} \frac{-\overline{a}_{il}}{\overline{a}_{ij}} x_l \\ x_k &= \overline{b}_k - \overline{a}_{kj} \overline{b}_i / \overline{a}_{ij} + \frac{\overline{a}_{kj}}{\overline{a}_{ij}} x_i + \sum_{l \in N \setminus j} (\overline{a}_{kl} - \frac{\overline{a}_{kj} \overline{a}_{il}}{\overline{a}_{ij}}) x_l \quad \text{for } k \in B \setminus i \end{aligned}$$

and the objective function expressed in the new basis becomes

$$z = \overline{z} - \frac{\overline{c}_j \overline{b}_i}{\overline{a}_{ij}} + \overline{c}_j x_i + \sum_{l \in N \setminus j} (\overline{c}_l - \frac{\overline{c}_j \overline{a}_{il}}{\overline{a}_{ij}}) x_l$$

We want the new basis to be dual feasible, i.e., we want

$$\overline{c}_j \leq 0 \quad \text{and} \quad \overline{c}_l - \frac{\overline{c}_j \overline{a}_{il}}{\overline{a}_{ij}} \leq 0 \quad \text{for all } l \in N \setminus j \; .$$

The first condition holds since $\overline{c}_j \leq 0$ by the fact that B is dual feasible. For the second condition, we observe that $\overline{c}_l \leq 0$, $\overline{c}_j \leq 0$ and $\overline{a}_{ij} > 0$ and hence the condition is certainly satisfied if $\overline{a}_{il} \leq 0$. So assume $\overline{a}_{il} > 0$. Then the condition is equivalent to

$$\frac{\overline{c}_l}{\overline{a}_{il}} - \frac{\overline{c}_j}{\overline{a}_{ij}} \leq 0 \; .$$

It is now clear how we have to choose j.

$$\text{Choose } j \in N \text{ such that } \overline{a}_{ij} > 0 \text{ and } \frac{\overline{c}_j}{\overline{a}_{ij}} = \max \left\{ \frac{\overline{c}_l}{\overline{a}_{il}} \; ; l \in N \text{ and } \overline{a}_{il} > 0 \right\} \; .$$

This completes the description of phase II of the dual simplex method.

How do we find an initial dual feasible basis? We describe two methods.

Consider

$$\text{(LP1)} \qquad \max \left\{ c^T x \; ; Ax = 0, \; x \in \mathbb{R}^n_{\geq 0} \right\} \; .$$

LP1 is feasible, since $x^* = 0$ is a feasible solution. Also if x^* is a feasible solution with $c^T x^* > 0$, LP1 is unbounded since tx^* with $t \in \mathbb{R}_{\geq 0}$ gives objective value $tc^T x^*$. We conclude that LP1 is either unbounded or has optimal objective value zero. Any basis B is feasible for LP1(since the right hand side is zero) and hence we may start Phase I of the primal simplex algorithm with an arbitrary basis. Phase I will either terminate with an optimal basis or declare the problem unbounded.

An optimal basis for LP1 is dual feasible for LP.

Assume now that LP1 is unbounded. Then LP is either infeasible or unbounded. Assume that LP is feasible, i.e., there is an x^* with $Ax^* = b$ and $x^* \geq 0$. Since LP1 is unbounded there is an $x' \geq 0$ with $Ax' = 0$ and $c^T x' > 0$. Consider $x(t) = x^* + tx'$ for arbitrary positive real number t: $x(t)$ is feasible and the objective values grows as a function of t. Thus LP is unbounded.

The decision whether LP is feasible can again be made by running phase I of the simplex method. This time on the problem $\max \left\{ 0^T x \; ; Ax = b, \; x \in \mathbb{R}^n_{\geq 0} \right\}$.

Here is another way. Let $M \in \mathbb{R}_{\geq 0}$ be such that every basic feasible solution of LP is contained in the box with $0 \leq x_i \leq M$ for all i. M can be deduced from Hadamard's bound on determinants of matrices and Cramer's rule. We start with

$$\text{(LP2)} \qquad \max \left\{ c^T x \; ; x_i + s_i = M, \; x_i, s_i \in \mathbb{R}_{\geq 0} \text{ for all } i \right\}$$

where s_i is a slack variable. The optimal solution to this LP is readily determined. We put x_i into the basis if $c_i > 0$ and we put s_i into the basis if $c_i \leq 0$. Let (x^*, s^*) be the optimal solution obtained in this way. We next consider the equalities $a_j^T x = b_j$ of LP. For each one we introduce a non-negative slack variable t_j, write $a_j^T x + t_j = b_j$ if $a_j^T x^* \geq b_j$ and $a_j^T - t_j = b_j$ if $a_j^T x^* < b_j$, and add t_j to the basis. In this way, we obtain a basic solution in which the t_j are non-positive. The basic solution is dual feasible. We are ready for phase II of the dual simplex algorithm. It either declares the problem infeasible or determines an optimal basic solution. If all s-variables (the slacks in the constraints $x_i + s_i = M$) are basic we simply drop these constraints and the s-variables and have an optimal solution of LP. If one of the s-variables is non-basic, LP is unbounded (why??).

1.6 Sensitivity Analysis

How much does the optimal objective value of an LP change after a modification of the right hand side? The dual is of great help to answer this question.

Consider $\max\left\{c^T x\,;Ax \leq b,\ x \in \mathbb{R}^n_{\geq 0}\right\}$, let δ^* be the optimal objective value, and let y^* be an optimal solution of the dual linear program $\min\left\{y^T b\,;y^T A \geq x, y \in \mathbb{R}^m_{\geq 0}\right\}$. Assume now that we replace the right hand side by $b+b'$ for some vector b'. The dual changes to $\min\left\{y^T(b+b')\,;y^T A \geq x, y \in \mathbb{R}^m_{\geq 0}\right\}$ and y^* stays a feasible solution of the dual. We conclude that the objective value of the modified primal is bounded above by

$$(y^*)^T(b+b') = (y^*)^T b + (y^*)^T b' = \delta^* + (y^*)^T b' \;.$$

If the basis corresponding to x^* is still feasible for the modified right hand side the estimate is sharp.

1.7 Reoptimization

Assume that we have determined an optimal solution to an LP. Can we use the solution to determine the solution to a modified LP? There are many ways to modify an LP. We may change the right hand side, we may change the cost function, we may add additional variables, or we may add additional constraints. We will see that in each of the four cases the optimal basis of the current LP can be used as a starting basis for the modified LP for either the primal or the dual simplex algorithm.

Let B be an optimal basis for $\max\left\{c^T x\,;Ax = b,\ x \in \mathbb{R}^n_{\geq 0}\right\}$ and let x^* be the corresponding optimal solution, i.e., $x_N^* = 0$ and $x_B^* = A_B^{-1} b$. We have $A_B^{-1} b \geq 0$ since B is primal feasible and we have $c_N^T - c_B^T A_B^{-1} A_N \leq 0$ since B is dual feasible. The nice fact about the optimality conditions is that one involves the right hand side and one involves the cost function, but neither involves both, and hence one of the optimality conditions stays true after a change of either the right hand side or the cost function.

Assume that we change the right hand side from b to b'. Then B stays dual feasible and hence we can take B as the initial basis for phase II of the dual simplex algorithm.

Assume next that we change the cost function from c to c'. Then B stays primal feasible and hence we can take B as the initial basis for phase II of the primal simplex algorithm.

We may also add an inequality $a^T x \leq d$. We introduce a slack variable s, write $a^T x + s = d$, make the coefficient of s in the cost function equal to zero, and add s to the basis. If x^* satisfies the inequality, the basis $B \cup s$ is optimal. If x^* does not satisfy the inequality, the basis $B \cup s$ is dual feasible (but not primal feasible) and hence we can use it as the initial basis for phase II of the dual simplex algorithm. The case of adding an equality is handled similarly.

We may also add a new variable x_{n+1} with column A_{n+1} in the constraint matrix and coefficient c_{n+1} in the objective function. We add x_{n+1} to the non-basic variables and compute its reduced cost as $c_{n+1} - c_B^T A_B^{-1} A_{N+1}$. B is still a feasible basis. If the reduced cost of x_{n+1} is non-positive, B is still optimal. If the reduced cost is positive, B is taken as the initial basis of phase II of the primal simplex method.

1.8 Totally Unimodular Constraint Matrices

Throughout this section we consider only LPs where right hand side and constraint matrix are integral.

We turn some more of the miracles mentioned in the introduction into theorems.

A $m \times n$ matrix A with $m \leq n$ is called unimodular, if any $n \times n$ submatrix of A has determinant $+1$, 0, or -1. If A is unimodular then the inverse of any $n \times n$ submatrix of A is integral. This is a consequence of Cramer's rule.

Theorem 13 *If the linear program* $\max\left\{c^T x \; ; Ax = b, x \in \mathbb{R}^n_{\geq 0}\right\}$ *is feasible and bounded and A is unimodular, the LP has an integral optimal solution.*

Proof: The non-basic variables are zero and hence integral. The value of the basic variables is given as $A_B^{-1} b$ and A_B^{-1} is integral. ∎

We also need to deal with linear programs in inequality form. A matrix A is called totally unimodular if any square submatrix of A has determinant $+1$, 0 or -1.

Lemma 4 *Let A be $m \times n$ and let I be an $m \times m$ identity matrix. Then (A, I) is unimodular.*

Proof: A square $m \times m$ submatrix of (A, I) selects some columns of A and some columns of I. The determinant of the matrix is equal to the determinant of a square submatrix of A. The claim follows. ∎

Theorem 14 *If the linear program* $\max\left\{c^T x \; ; Ax \leq b, x \in \mathbb{R}^n_{\geq 0}\right\}$ *is feasible and bounded and A is totally unimodular, the LP has an integral optimal solution.*

Proof: After the addition of slack variables we obtain $\max\left\{c^T x \; ; Ax + Is = b, x \in \mathbb{R}^n_{\geq 0}, s \in \mathbb{R}^s_{\geq 0}\right\}$. The matrix (A, I) is unimodular by the preceding lemma. ∎

We next give some applications.

Directed Node-Edge Incidence Matrices of Directed Graphs: Consider a directed graph $G = (V, E)$. The directed node-edge incidence matrix has $|V|$ rows and $|E|$ columns. For each edge $e = (u, v)$ we have $+1$ in column e and row u and -1 in row v and column e.

Lemma 5 *The directed node-edge incidence matrix of directed graphs is totally unimodular.*

Proof: A square submatrix M is determined by a set U of nodes and a set A of arcs; U and A have the same cardinality since our submatrix is square. If there is an edge in A with no endpoint in U, M has a zero-column and hence its determinant is zero. Thus we may assume that every edge in A has at least one endpoint in U. If there is an edge $e \in A$ having exactly one endpoint in U, say u, we develop the matrix according to the column e. Since M has exactly one non-zero entry in column e and this entry is ± 1, the determinant of M is ± 1 times the determinant of the matrix determined by $U \setminus u$ and $A \setminus e$. Continuing in this way, we will either reduce the matrix to a 1×1 matrix and then the claim is obvious or we will arrive at the situation where every arc in A has both endpoints in U. In this case, adding all rows of A gives us the zero vector (since every column contains a $+1$ and a -1) and hence A is singular. ∎

Corollary 15 *The maximum flow problem with integer capacities has an integral optimal solution. The minimum cost flow problem with integer capacities has an integral optimal solution.*

Proof: There are two type of constraints, the mass balance constraints and the capacity constraints. We have a mass balance constraint for each node (in the maximum flow problem for each node except the source and the sink). The balance constraint for node v reads

$$\sum_{e \in Out(v)} x_e = b(v) + \sum_{e \in In(v)} x_e$$

where x_e is the flow on e and $b(v)$ is the supply/demand at node v. In the maximum flow problem, $b(v) = 0$ for all v. Rearranging, we obtain

$$\sum_{e \in Out(v)} x_e - \sum_{e \in In(v)} x_e = b(v) \ ,$$

In the row for node v, the edges out of v have entry $+1$ and the edges into v have entry -1. We also have a capacity constraint for each edge

$$x_e \leq cap(e)$$

for each edge (in the uncapacitated problem there are no capacity constraints).

We conclude that our constraint matrix has $n + m$ rows, one for each node and one for each edge, and m columns, one for each edge. The first n rows form a directed node-edge incident matrix and the last m rows form an identity matrix. The (absolute value) of the determinant of any square submatrix is therefor equal to the determinant of a square submatrix of the node-edge incident matrix. The claim follows.

∎

Undirected Node-Edge Incidence Matrices of Bipartite Undirected Graphs: Consider an undirected graph $G = (V, E)$. The undirected node-edge incidence matrix has $|V|$ rows and $|E|$ columns. For each edge $e = (u, v)$ we have a $+1$ in column e and rows u and v.

Lemma 6 *The node-edge incidence matrix of undirected bipartite graphs is totally unimodular.*

Proof: If G is bipartite we may split the node set V into $A \cup B$ such that every edge has exactly one endpoint in A and exactly one endpoint in B. We turn the graph into a directed graph by directing all edges form A to B. We obtain the directed (!!!) node-edge incidence matrix graph of the directed graph by multiplying the entries in the rows corresponding to nodes in B by -1. This may change the sign of the determinant. We now have the node-edge incidence matrix of a directed graph which is totally unimodular by the preceding lemma. ∎

The following integer (!!!) linear program defines the maximum weight matching problem in graphs. We have a 0-1 variable x_e for every edge and a cost c_e for every edge. The goal is to maximize $\sum_e c_e x_e$ subject to $\sum_{e \in \delta(v)} \leq 1$ for every node v. The constraint matrix of this problem is an undirected node-edge incident matrix. For bipartite graphs the constraint matrix is totally unimodular and hence we may drop the constraint $x_e \in \{0, 1\}$. The resulting LP is guaranteed to have an integral optimal solution.

Corollary 16 *The maximum weight matching LP in bipartite graphs has an integral solution.*

Proof: The constraint matrix is the node-edge incidence matrix of a bipartite undirected graph. ∎

Warning: For general graphs the undirected node-edge incident matrix is not totally unimodular. Indeed, consider a cycle on three nodes. Its node-edge incident matrix is

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

and has determinant equal to two. Assume that we give all edges of the cycle cost equal to one. Then a maximum weight matching consists of one edge and hence has cost one. On the other hand, an optimal solution to the LP sets $x_e = 1/2$ for every edge e and hence has value $3/2$.

1.9 Primal-Dual Combinatorial Algorithms

1.10 Running Time of Simplex Method

discuss Klee-Minty cube

discuss average case results by Borgward and Spielmann

discuss randomized simplex by Seidel and by Matousek, Sharir, Welzl.

1.11 Alternative Algorithms for Solving Linear Programs