MATTHIAS GERDTS

LINEAR PROGRAMMING
MSM 2Da (G05a,M09a)

Address of the Author:

Matthias Gerdts

Management Mathematics Group
School of Mathematics
University of Birmingham
Edgbaston
Birmingham B15 2TT

E-Mail: gerdtsm@maths.bham.ac.uk
WWW: web.mat.bham.ac.uk/M.Gerdts

Preliminary Version: December 4, 2008

Contents

1 Introduction **2**

1.1 Examples . . . 7

1.2 Transformation to canonical and standard form . . . 10

1.3 Graphical Solution of LP . . . 17

1.4 The Fundamental Theorem of Linear Programming . . . 23

1.5 Software for LP's . . . 26

2 The Simplex Method **27**

2.1 Vertices and Basic Solutions . . . 28

2.2 The (primal) Simplex Method . . . 30

2.2.1 Computing a neighbouring feasible basic solution . . . 34

2.2.2 The Algorithm . . . 39

2.3 The Simplex Table . . . 42

2.4 Phase 1 of the Simplex Method . . . 47

2.5 Computational Issues . . . 51

2.5.1 Finite Termination of the Simplex Method . . . 51

2.5.2 Complexity of the Simplex Method . . . 55

2.5.3 The Revised Simplex Method . . . 57

3 Duality and Sensitivity **62**

3.1 The Dual Linear Program . . . 62

3.2 Weak and Strong Duality . . . 66

3.3 Dual Simplex Method . . . 68

3.4 Sensitivities and Shadow Prices . . . 72

Bibliography **81**

Lecture plan

Lectures:

Date	Hours	Pages
01.10.2007	9:00-9:50	1–6
02.10.2007	9:00-9:50	7–8
08.10.2007	9:00-9:50	9–13
09.10.2007	9:00-9:50	14–18
15.10.2007	9:00-9:50	18–21
16.10.2007	9:00-9:50	22–25
22.10.2007	9:00-9:50	25–28
23.10.2007	9:00-9:50	29–32
29.10.2007	9:00-9:50	32–35
30.10.2007	9:00-9:50	35–40
05.11.2007	9:00-9:50	40–42
06.11.2007	9:00-9:50	CT
12.11.2007	9:00-9:50	43–48
13.11.2007	9:00-9:50	48–53
19.11.2007	9:00-9:50	54–57
20.11.2007	9:00-9:50	57–61
26.11.2007	9:00-9:50	62–66
27.11.2007	9:00-9:50	66–68
03.12.2007	9:00-9:50	68–72
04.12.2007	9:00-9:50	CT
10.12.2007	9:00-9:50	72–76
11.12.2007	9:00-9:50	76–80

Examples classes:

Date	Hours	Questions
13.10.2008	14:00-14:50	
27.10.2008	14:00-14:50	
10.11.2008	14:00-14:50	
24.11.2008	14:00-14:50	
08.12.2008	14:00-14:50	

Class tests:

Date	Hours	Questions
06.11.2008	9:00-9:50	
04.12.2008	9:00-9:50	

Chapter 1

Introduction

This course is about linear optimisation which is also known as linear programming. Linear programming is an important field of optimisation. Many practical problems in operations research can be expressed as linear programming problems. A number of algorithms for other types of optimisation problems work by solving linear programming problems as sub-problems. Historically, ideas from linear programming have inspired many of the central concepts of optimisation theory, such as duality, decomposition, and the importance of convexity and its generalisations. Likewise, linear programming is heavily used in business management, either to maximize the income or minimize the costs of a production scheme.

What is optimisation about?

Optimisation plays a major role in many applications from economics, operations research, industry, engineering sciences, and natural sciences. In the famous book of Nocedal and Wright [NW99] we find the following statement:

> People optimise: Airline companies schedule crews and aircraft to minimise cost. Investors seek to create portfolios that avoid risks while achieving a high rate of return. Manufacturers aim for maximising efficiency in the design and operation of their production processes.
> Nature optimises: Physical systems tend to a state of minimum energy. The molecules in an isolated chemical system react with each other until the total potential energy of their electrons is minimised, Rays of light follow paths that minimise their travel time.

General optimisation problem:
For a given function $f : \mathbb{R}^n \to \mathbb{R}$ and a given set $M \subseteq \mathbb{R}^n$ find $\hat{x} \in M$ which minimises f on M, that is $\hat{x}$ satisfies

$$f(\hat{x}) \leq f(x) \qquad \text{for every } x \in M.$$

Terminology:

- f is called objective function.

- M is called feasible set (or admissible set).
- $\hat{x}$ is called minimiser of f on M.
- Any $x \in M$ is called feasible or admissible.
- The components x_i, $i = 1, \ldots, n$, of the vector x are called optimisation variables. For brevity we likewise call the vector x optimisation variable or simply variable.

Remarks:

- Without loss of generality, it suffices to consider minimisation problems. For, if the task were to maximise f, an equivalent minimisation problem is given by minimising $-f$. [Exercise: Define what equivalent means in this context!]
- The function f often is associated with costs.
- The set M is often associated with resources, e.g. available budget, available material in a production, etc.

The above general optimisation problem is much too general to solve it and one needs to specify f and M. In this course we exclusively deal with linear optimisation problems.

What is a linear function?

Definition 1.0.1 (linear function)
A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is called linear, if and only if

$$\begin{aligned} f(x+y) &= f(x) + f(y), \\ f(cx) &= cf(x) \end{aligned}$$

hold for every $x, y \in \mathbb{R}^n$ and every $c \in \mathbb{R}$.

Linear vs. nonlinear functions:

Example 1.0.2
The functions

$$\begin{aligned} f_1(x) &= x, \\ f_2(x) &= 3x_1 - 5x_2, \qquad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \\ f_3(x) &= c^\top x, \qquad x, c \in \mathbb{R}^n, \\ f_4(x) &= Ax, \qquad x \in \mathbb{R}^n,\ A \in \mathbb{R}^{m\times n}, \end{aligned}$$

are linear [exercise: prove it!].
The functions

$$\begin{aligned} f_5(x) &= 1, \\ f_6(x) &= x+1, \qquad x \in \mathbb{R}, \\ f_7(x) &= Ax+b, \qquad x \in \mathbb{R}^n,\ A \in \mathbb{R}^{m\times n},\ b \in \mathbb{R}^m, \\ f_8(x) &= x^2, \\ f_9(x) &= \sin(x), \\ f_{10}(x) &= a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0, \quad a_i \in \mathbb{R},\ (a_n,\ldots,a_2,a_0) \neq 0 \end{aligned}$$

are not linear [exercise: prove it!].

Fact:
Every linear function $f : \mathbb{R}^n \to \mathbb{R}^m$ can be expressed in the form $f(x) = Ax$ with some matrix $A \in \mathbb{R}^{m\times n}$.

Notion:
Throughout this course we use the following notion:

- $x \in \mathbb{R}^n$ is the column vector $\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$ with components $x_1, x_2, \ldots, x_n \in \mathbb{R}$.
- By $x^\top$ we mean the row vector $(x_1, x_2, \ldots, x_n)$. This is called the transposed vector of x.
- $A \in \mathbb{R}^{m\times n}$ is the matrix $\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$ with entries a_{ij}, $1 \leq i \leq m$, $1 \leq j \leq n$.
- By $A^\top \in \mathbb{R}^{n\times m}$ we mean the transposed matrix of A, i.e. $\begin{pmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{pmatrix}$.

This notion allows to characterise linear optimisation problems. A linear optimisation problem is a general optimisation problem in which f is a linear function and the set M is defined by the intersection of finitely many linear equality or inequality constraints. A linear program in its most general form has the following structure:

Definition 1.0.3 **(Linear Program (LP))**
Minimise

$$c^\top x = c_1x_1 + \ldots + c_nx_n = \sum_{i=1}^{n} c_ix_i$$

subject to the constraints

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \quad \left\{ \begin{array}{c} \leq \\ \geq \\ = \end{array} \right\} \quad b_i, \qquad i = 1,\ldots,m.$$

In addition, some (or all) components x_i*,* $i \in I$*, where* $I \subseteq \{1,\ldots,n\}$ *is an index set, may be restricted by sign conditions according to*

$$x_i \left\{ \begin{array}{c} \leq \\ \geq \end{array} \right\} 0, \qquad i \in I.$$

Herein, the quantities

$$c = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} \in \mathbb{R}^n, \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \in \mathbb{R}^m, \quad A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \in \mathbb{R}^{m\times n}$$

are given data.

Remark 1.0.4

- *Without loss of generality we restrict the discussion to minimisation problems. Notice, that every maximisation problem can be transformed easily into an equivalent minimisation problem by multiplying the objective function with* -1*, i.e. maximisation of* $c^\top x$ *is equivalent to minimising* $-c^\top x$ *and vice versa.*
- *The relations '*$\leq$*', '*$\geq$*', and '*$=$*' are applied component-wise to vectors, i.e. given two vectors* $x = (x_1,\ldots,x_n)^\top$ *and* $y = (y_1,\ldots,y_n)^\top$ *we have*

$$x \left\{ \begin{array}{c} \leq \\ \geq \\ = \end{array} \right\} y \quad \Leftrightarrow \quad x_i \left\{ \begin{array}{c} \leq \\ \geq \\ = \end{array} \right\} y_i, \quad i = 1,\ldots,n.$$

- *Variables* x_i*,* $i \notin I$*, are called free variables, as these variables may assume any real value.*

Example 1.0.5

The LP

$$\begin{array}{lrcrcrcrcr} \textit{Minimise} & -5x_1 & + & x_2 & + & 2x_3 & + & 2x_4 & & \\ \textit{s.t.} & & & x_2 & - & 4x_3 & & & \leq & 5 \\ & 2x_1 & + & x_2 & - & x_3 & - & x_4 & \geq & -3 \\ & x_1 & + & 3x_2 & - & 4x_3 & - & x_4 & = & -1 \\ & x_1 \leq 0 & , & x_2 \geq 0 & , & & & x_4 \geq 0 & & \end{array}$$

is a general LP with $I = \{1, 2, 4\}$ *and*

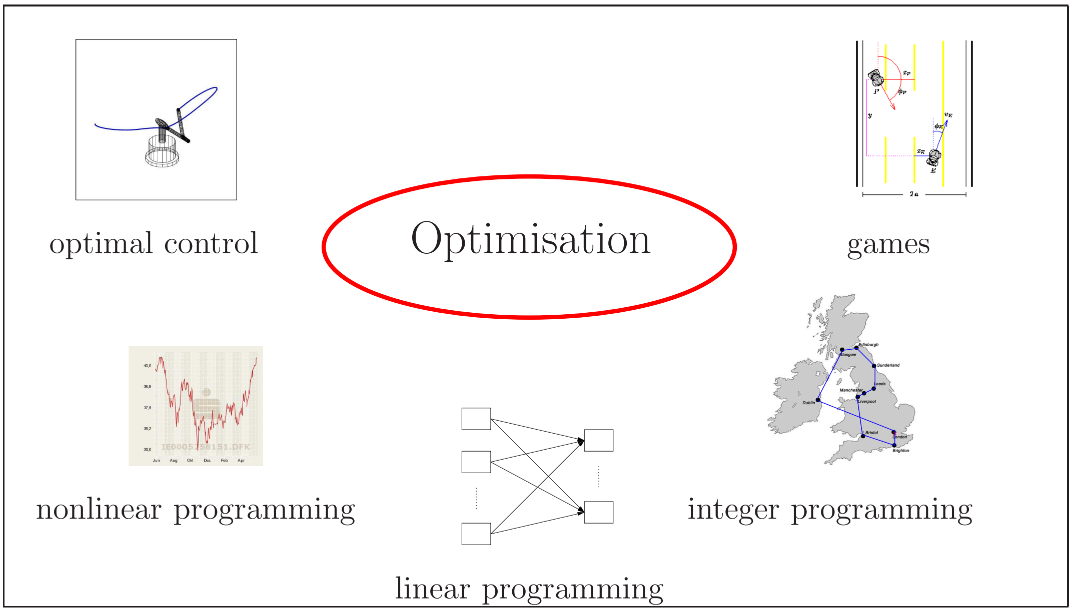
$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}, \; c = \begin{pmatrix} -5 \\ 1 \\ 2 \\ 2 \end{pmatrix}, \; b = \begin{pmatrix} 5 \\ -3 \\ -1 \end{pmatrix}, \; A = \begin{pmatrix} 0 & 1 & -4 & 0 \\ 2 & 1 & -1 & -1 \\ 1 & 3 & -4 & -1 \end{pmatrix}.$$

The variable x_3 *is free.*

The following questions arising in the context of linear programming will be investigated throughout this course:

- Existence of feasible points?
- Existence of optimal solutions?
- Uniqueness of an optimal solution?
- How does an optimal solution depend on problem data?
- Which algorithms can be used to compute an optimal solution?
- Which properties do these algorithms possess (finiteness, complexity)?

Beyond linear programming ...



Typical applications (there are many more!):

- nonlinear programming: portfolio optimisation, shape optimisation, parameter identification, topology optimisation, classification
- linear programming: allocation of resources, transportation problems, transshipment problems, maximum flows
- integer programming: assignment problems, travelling salesman problems, VLSI design, matchings, scheduling, shortest paths, telecommunication networks, public transportation networks
- games: economical behaviour, equilibrium problems, electricity markets
- optimal control: path planning for robots, cars, flight systems, crystal growth, flows, cooling of steel, economical strategies, inventory problems

1.1 Examples

Before defining linear programs precisely, we will discuss some typical examples.

Example 1.1.1 **(Maximisation of Profit)**
A farmer intends to plant 40 acres with sugar beets and wheat. He can use 2400 pounds and 312 working days to achieve this. For each acre his cultivation costs amount to 40 pounds for sugar beets and to 120 pounds for wheat. For sugar beets he needs 6 working days per acre and for wheat 12 working days per acre. The profit amounts to 100 pounds per acre for sugar beets and to 250 pounds per acre for wheat. Of course, the farmer wants to maximise his profit.

Mathematical formulation: Let x_1 denote the acreage which is used for sugar beets and x_2 those for wheat. Then, the profit amounts to

$$f(x_1, x_2) = 100x_1 + 250x_2.$$

The following restrictions apply:

$$\begin{array}{ll} \textit{maximum size:} & x_1 + x_2 \le 40 \\ \textit{money:} & 40x_1 + 120x_2 \le 2400 \\ \textit{working days:} & 6x_1 + 12x_2 \le 312 \\ \textit{no negative acreage:} & x_1, x_2 \ge 0 \end{array}$$

In matrix notation we obtain

$$\max \quad c^\top x \quad s.t. \quad Ax \le b, \; x \ge 0,$$

where

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad c = \begin{pmatrix} 100 \\ 250 \end{pmatrix}, \quad b = \begin{pmatrix} 40 \\ 2400 \\ 312 \end{pmatrix}, \quad A = \begin{pmatrix} 1 & 1 \\ 40 & 120 \\ 6 & 12 \end{pmatrix}.$$

Example 1.1.2 **(The Diet Problem by G.J.Stigler, 1940ies)**
A human being needs vitamins S1, S2, and S3 for healthy living. Currently, only 4 medications A1,A2,A3,A4 contain these substances:

	S1	*S2*	*S3*	*cost*
A1	*30*	*10*	*50*	*1500*
A2	*5*	*0*	*3*	*200*
A3	*20*	*10*	*50*	*1200*
A4	*10*	*20*	*30*	*900*
need per day	*10*	*5*	*5.5*	

Task: Find combination of medications that satisfy the need at minimal cost!
Let x_i denote the amount of medication A_i, $i = 1, 2, 3, 4$. The following linear program solves the task:

$$\begin{array}{lrrrrcl} \textit{Minimise} & \multicolumn{4}{l}{1500x_1+200x_2+1200x_3+900x_4} & & \\ \textit{subject to} & 30x_1+ & 5x_2+ & 20x_2+ & 10x_4 & \ge & 10 \\ & 10x_1+ & & 10x_3+ & 20x_4 & \ge & 5 \\ & 50x_1+ & 3x_2+ & 50x_3+ & 30x_4 & \ge & 5.5 \\ & \multicolumn{4}{l}{x_1, x_2, x_3, x_4 \ge 0} & & \end{array}$$

In matrix notation, we obtain

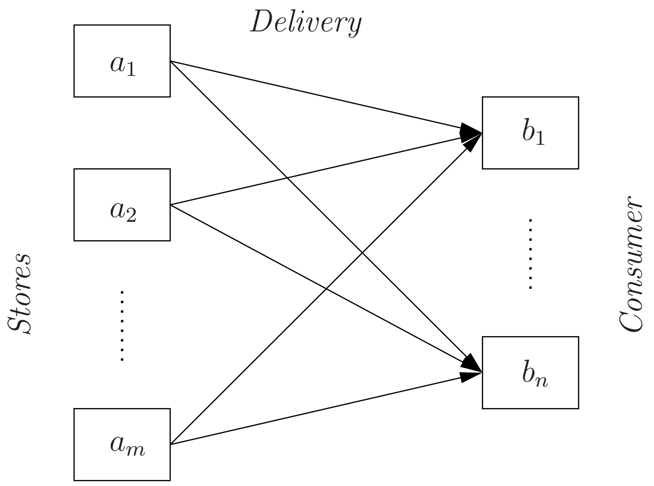
$$\min \quad c^\top x \quad s.t. \quad Ax \ge b, \; x \ge 0,$$

where

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}, \quad c = \begin{pmatrix} 1500 \\ 200 \\ 1200 \\ 900 \end{pmatrix}, \quad b = \begin{pmatrix} 10 \\ 5 \\ 5.5 \end{pmatrix}, \quad A = \begin{pmatrix} 30 & 5 & 20 & 10 \\ 10 & 0 & 10 & 20 \\ 50 & 3 & 50 & 30 \end{pmatrix}.$$

In the 1940ies nine people spent in total 120 days (!) for the numerical solution of a diet problem with 9 inequalities and 77 variables.

Example 1.1.3 **(Transportation problem)**
A transport company has m stores and wants to deliver a product from these stores to n consumers. The delivery of one item of the product from store i to consumer j costs c_{ij} pound. Store i has stored a_i items of the product. Consumer j has a demand of b_j items of the product. Of course, the company wants to satisfy the demand of all consumers. On the other hand, the company aims at minimising the delivery costs.

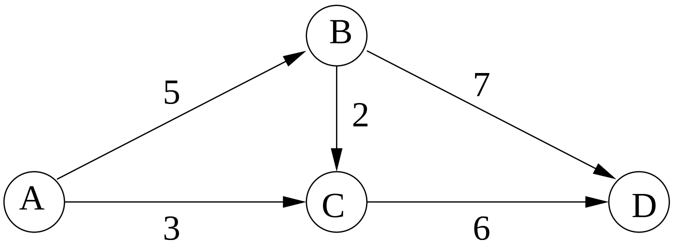


Let x_{ij} denote the amount of products which are delivered from store i to consumer j. In order to find the optimal transport plan, the company has to solve the following linear program:

$$\begin{array}{llcll}
\textit{Minimise} & \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} & & & \textit{(minimise delivery costs)} \\
\textit{s.t.} & \displaystyle\sum_{j=1}^{n} x_{ij} & \leq & a_i, \quad i = 1,\ldots,m, & \textit{(can't deliver more than it is there)} \\
& \displaystyle\sum_{i=1}^{m} x_{ij} & \geq & b_j, \quad j = 1,\ldots,n, & \textit{(satisfy demand)} \\
& x_{ij} & \geq & 0, \quad i = 1,\ldots,m,\ j = 1,\ldots,n. & \textit{(can't deliver negative amount)}
\end{array}$$

Remark: *In practical problems x_{ij} is often restricted in addition by the constraint $x_{ij} \in \mathbb{N}_0 = \{0, 1, 2, \ldots\}$, i.e. x_{ij} may only assume integer values. This restriction makes the problem much more complicated and standard methods like the simplex method cannot be applied anymore.*

Example 1.1.4 **(Network problem)**
A company intends to transport as many goods as possible from city A to city D on the road network below. The figure next to each edge in the network denotes the maximum capacity of the edge.



How can we model this problem as a linear program?
Let $V = \{A, B, C, D\}$ denote the nodes of the network (they correspond to cities). Let $E = \{(A, B), (A, C), (B, C), (B, D), (C, D)\}$ denote the edges of the network (they correspond to roads connecting two cities).
For each edge $(i, j) \in E$ let x_{ij} denote the actual amount of goods transported on edge (i, j) and u_{ij} the maximum capacity. The capacity constraints hold:

$$0 \leq x_{ij} \leq u_{ij} \qquad \forall (i, j) \in E.$$

Moreover, it is reasonable to assume that no goods are lost or no goods are added in cities B and C, i.e. the conservation equations 'outflow - inflow = 0' hold:

$$\begin{aligned} x_{BD} + x_{BC} - x_{AB} &= 0, \\ x_{CD} - x_{AC} - x_{BC} &= 0. \end{aligned}$$

The task is to maximize the amount of goods leaving city A (note that this is the same amount that enters city D):

$$x_{AB} + x_{AC} \to \max.$$

All examples are special cases of the general LP 1.0.3.

1.2 Transformation to canonical and standard form

LP in its most general form is rather tedious for the construction of algorithms or the derivation of theoretical properties. Therefore, it is desirable to restrict the discussion to certain standard forms of LP into which any arbitrarily structured LP can be transformed.

In the sequel we will restrict the discussion to two types of LP: the canonical LP and the standard LP.
We start with the canonical LP.

Definition 1.2.1 **(Canonical Linear Program** ($\mathrm{LP_C}$)**)**
Let $c = (c_1, \ldots, c_n)^\top \in \mathbb{R}^n$, $b = (b_1, \ldots, b_m)^\top \in \mathbb{R}^m$ *and*

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \in \mathbb{R}^{m\times n}$$

be given. The canonical linear program reads as follows: Find $x = (x_1, \ldots, x_n)^\top \in \mathbb{R}^n$ *such that the objective function*

$$\sum_{i=1}^{n} c_i x_i$$

becomes minimal subject to the constraints

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i, \qquad i = 1, \ldots, m, \tag{1.1}$$

$$x_j \geq 0, \qquad j = 1, \ldots, n. \tag{1.2}$$

In matrix notation:

$$\text{Minimise} \quad c^\top x \quad \text{subject to} \quad Ax \leq b, \; x \geq 0. \tag{1.3}$$

We need some notation.

Definition 1.2.2 **(objective function, feasible set, feasible points, optimality)**

(i) The function $f(x) = c^\top x$ *is called objective function.*

(ii) The set

$$M_C := \{x \in \mathbb{R}^n \mid Ax \leq b, \; x \geq 0\}$$

is called feasible (or admissible) set (of $\mathrm{LP_C}$*).*

(iii) A vector $x \in M_C$ *is called feasible (or admissible) (for* $\mathrm{LP_C}$*).*

(iv) $\hat{x} \in M_C$ *is called optimal (for* $\mathrm{LP_C}$*), if*

$$c^\top \hat{x} \leq c^\top x \quad \forall x \in M_C.$$

We now discuss how an arbitrarily structured LP can be transformed into LP_{C}. The following cases may occur:

(i) Inequality constraints: The constraint

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i$$

can be transformed into a constraint of type (1.1) by multiplication with -1:

$$\sum_{j=1}^{n} (-a_{ij}) x_j \leq -b_i.$$

(ii) Equality constraints: The constraint

$$\sum_{j=1}^{n} a_{ij} x_j = b_i$$

can be written equivalently as

$$b_i \leq \sum_{j=1}^{n} a_{ij} x_j \leq b_i.$$

Hence, instead of one equality constraint we obtain two inequality constraints. Using (i) for the inequality on the left we finally obtain two inequality constraints of type (1.1):

$$\begin{aligned} \sum_{j=1}^{n} a_{ij} x_j &\leq b_i, \\ \sum_{j=1}^{n} (-a_{ij}) x_j &\leq -b_i. \end{aligned}$$

(iii) Free variables: Any real number x_i can be decomposed into $x_i = x_i^+ - x_i^-$ with nonnegative numbers $x_i^+ \geq 0$ and $x_i^- \geq 0$. Notice, that this decomposition is not unique as for instance $-6 = 0 - 6 = 1 - 7 = \ldots$. It will be unique if we postulate that either x_i^+ or x_i^- be zero. But this postulation is not a linear constraint anymore, so we don't postulate it.

In order to transform a given LP into canonical form, every occurrence of the free variable x_i in LP has to be replaced by $x_i^+ - x_i^-$. The constraints $x_i^+ \geq 0$ and $x_i^- \geq 0$ have to be added. Notice, that instead of one variable x_i we now have two variables x_i^+ and x_i^-.

(iv) Maximisation problems: Maximising $c^\top x$ is equivalent to minimising $(-c)^\top x$.

Example 1.2.3

Consider the linear program of maximising

$$-1.2x_1 - 1.8x_2 - x_3$$

subject to the constraints

$$\begin{aligned} x_1 &\geq -\frac{1}{3} \\ x_1 - 2x_3 &\leq 0 \\ x_1 - 2x_2 &\leq 0 \\ x_2 - x_1 &\leq 0 \\ x_3 - 2x_2 &\leq 0 \\ x_1 + x_2 + x_3 &= 1 \\ x_2, x_3 &\geq 0. \end{aligned}$$

This linear program is to be transformed into canonical form (1.3).
The variable x_1 *is a free variable. Therefore, according to (iii) we replace it by* $x_1 := x_1^+ - x_1^-$, $x_1^+, x_1^- \geq 0$.
According to (ii) the equality constraint $x_1 + x_2 + x_3 = 1$ *is replaced by*

$$\underbrace{x_1^* - x_1^-}_{=x_1} + x_2 + x_3 \leq 1 \qquad \text{and} \qquad \underbrace{-x_1^+ + x_1^-}_{=-x_1} - x_2 - x_3 \leq -1.$$

According to (i) the inequality constraint $x_1 = x_1^+ - x_1^- \geq -\frac{1}{3}$ *is replaced by*

$$-x_1 = -x_1^+ + x_1^- \leq \frac{1}{3}.$$

Finally, we obtain a minimisation problem by multiplying the objective function with -1.
Summarising, we obtain the following canonical linear program:

$$\text{Minimise} \quad (1.2, -1.2, 1.8, 1) \begin{pmatrix} x_1^+ \\ x_1^- \\ x_2 \\ x_3 \end{pmatrix}$$

$$\text{s.t.} \quad \begin{pmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & -2 \\ 1 & -1 & -2 & 0 \\ -1 & 1 & 1 & 0 \\ 0 & 0 & -2 & 1 \\ 1 & -1 & 1 & 1 \\ -1 & 1 & -1 & -1 \end{pmatrix} \begin{pmatrix} x_1^+ \\ x_1^- \\ x_2 \\ x_3 \end{pmatrix} \leq \begin{pmatrix} 1/3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{pmatrix}, \qquad \begin{pmatrix} x_1^+ \\ x_1^- \\ x_2 \\ x_3 \end{pmatrix} \geq 0.$$

Theorem 1.2.4

Using the transformation techniques (i)–(iv), we can transform any LP into a canonical linear program (1.3).

The canonical form is particularly useful for visualising the constraints and solving the problem graphically. The feasible set M_C is an intersection of finitely many halfspaces and can be visualised easily for $n = 2$ (and maybe $n = 3$). This will be done in the next section. But for the construction of solution methods, in particular for the simplex method, another notation is preferred: the standard linear program.

Definition 1.2.5 **(Standard Linear Program ($\mathrm{LP_S}$))**

Let $c = (c_1, \dots, c_n)^\top \in \mathbb{R}^n$, $b = (b_1, \dots, b_m)^\top \in \mathbb{R}^m$ *and*

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \in \mathbb{R}^{m\times n}$$

be given. Let Rank$(A) = m$. *The standard linear program reads as follows: Find* $x = (x_1, \dots, x_n)^\top \in \mathbb{R}^n$ *such that the objective function*

$$\sum_{i=1}^{n} c_i x_i$$

becomes minimal subject to the constraints

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \qquad i = 1, \dots, m, \tag{1.4}$$

$$x_j \geq 0, \qquad j = 1, \dots, n. \tag{1.5}$$

In matrix notation:

$$\textit{Minimise} \quad c^\top x \quad \textit{subject to} \quad Ax = b,\ x \geq 0. \tag{1.6}$$

Remark 1.2.6

- *The sole difference between* $\mathrm{LP_C}$ *and* $\mathrm{LP_S}$ *are the constraints* (1.1) *and* (1.4)*, respectively. Don't be confused by the same notation. The quantities c, b and A are* **not** *the same when comparing* $\mathrm{LP_C}$ *and* $\mathrm{LP_S}$.

- *The rank assumption Rank$(A) = m$ is useful because it avoids linearly dependent constraints. Linearly dependent constraints always can be detected and eliminated in a preprocessing procedure.*

- LP_S *is only meaningful if $m < n$ holds. Because in the case $m = n$ and A nonsingular, the equality constraint yields $x = A^{-1}b$. Hence, x is completely defined and no degrees of freedom remain for optimising x. If $m > n$ and Rank$(A) = n$ the linear equation $Ax = b$ possesses at most one solution and again no degrees of freedom remain left for optimisation. Without stating it explicitely, we will assume $m < n$ in the sequel whenever we discuss* LP_S.

As in Definition 1.2.2 we define the feasible set (of LP_S) to be

$$M_S := \{x \in \mathbb{R}^n \mid Ax = b,\ x \geq 0\}$$

and call $\hat{x} \in M_S$ optimal (for LP_S) if

$$c^\top \hat{x} \leq c^\top x \quad \forall x \in M_S.$$

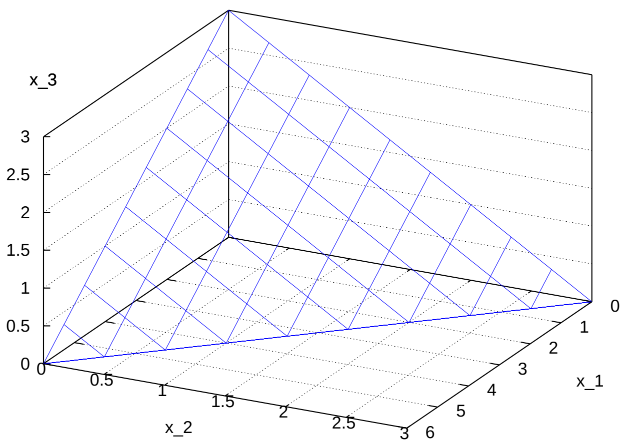
The feasible set M_S is the intersection of the affine subspace $\{x \in \mathbb{R} \mid Ax = b\}$ (e.g. a line or a plane) with the nonnegative orthant $\{x \in \mathbb{R}^n \mid x \geq 0\}$. The visualisation is more difficult as for the canonical linear program and a graphical solution is rather impossible (except for very simple cases).

Example 1.2.7

Consider the linear program

$$\max \quad -90x_1 - 150x_2 \quad s.t. \quad \frac{1}{2}x_1 + x_2 + x_3 = 3,\ x_1, x_2, x_3 \geq 0.$$

The feasible set $M_S = \{(x_1, x_2, x_3)^\top \in \mathbb{R}^3 \mid \frac{1}{2}x_1 + x_2 + x_3 = 3,\ x_1, x_2, x_3 \geq 0\}$ is the blue triangle below:



We can transform a canonical linear program into an equivalent standard linear program and vice versa.

(i) Transformation of a canonical problem into a standard problem:
Let a canonical linear program (1.3) be given.

Define slack variables $y = (y_1, \ldots, y_m)^\top \in \mathbb{R}^m$ by

$$y_i := b_i - \sum_{j=1}^{n} a_{ij} x_j, \qquad i = 1, \ldots, m,$$

resp.

$$y = b - Ax.$$

It holds $Ax \leq b$ if and only if $y \geq 0$.

With $\hat{x} := (x, y)^\top \in \mathbb{R}^{n+m}$, $\hat{c} := (c, 0)^\top \in \mathbb{R}^{n+m}$, $\hat{A} := (A \mid I) \in \mathbb{R}^{m\times(n+m)}$ we obtain a standard linear program of type (1.6):

$$\text{Minimise} \quad \hat{c}^\top \hat{x} \quad \text{s.t.} \quad \hat{A}\hat{x} = b,\ \hat{x} \geq 0. \tag{1.7}$$

Notice, that $\text{Rank}(\hat{A}) = m$ holds automatically.

(ii) Transformation of a standard problem into a canonical problem:
Let a standard linear program (1.6) be given. Rewrite the equality $Ax = b$ as two inequalities $Ax \leq b$ and $-Ax \leq -b$. With

$$\hat{A} := \begin{pmatrix} A \\ -A \end{pmatrix} \in \mathbb{R}^{2m\times n}, \qquad \hat{b} := \begin{pmatrix} b \\ -b \end{pmatrix} \in \mathbb{R}^{2m}$$

a canonical linear program of type (1.3) arises:

$$\text{Minimise} \quad c^\top x \quad \text{s.t.} \quad \hat{A}x \leq \hat{b},\ x \geq 0. \tag{1.8}$$

We obtain

Theorem 1.2.8

Canonical linear programs and standard linear programs are equivalent in the sense that

(i) x solves (1.3) if and only if $\hat{x} = (x, b - Ax)^\top$ solves (1.7).

(ii) x solves (1.6) if and only if x solves (1.8).

Proof: We only show (i) as (ii) is obvious.

(i) $\Rightarrow$ Let x solve (1.3), i.e. it holds

$$c^\top x \leq c^\top z \qquad \forall z \ : \ Az \leq b, \ z \geq 0.$$

Let $\hat{z} = (z, y)^\top$ be feasible for (1.7). Then, it holds $y = b - Az$ and $y \geq 0$ and hence, z satisfies $Az \leq b$ and $z \geq 0$. Then,

$$\hat{c}^\top \hat{x} = c^\top x \leq c^\top z = \hat{c}^\top \hat{z}.$$

The assertion follows as $\hat{z}$ was an arbitrary feasible point.

(i)$\Leftarrow$ Let $\hat{x} = (x, b - Ax)^\top$ solve (1.7), i.e.

$$\hat{c}^\top \hat{x} \leq \hat{c}^\top \hat{z} \qquad \forall \hat{z} \ : \ \hat{A}\hat{z} = b, \ \hat{z} \geq 0.$$

Let z be feasible for (1.3). Then $\hat{z} = (z, b - Az)^\top$ is feasible for (1.7) and

$$c^\top x = \hat{c}^\top \hat{x} \leq \hat{c}^\top \hat{z} = c^\top z.$$

The assertion follows as z was an arbitrary feasible point. □

1.3 Graphical Solution of LP

We demonstrate the graphical solution of a canonical linear program in the 2-dimensional space.

Example 1.3.1 **(Example 1.1.1 revisited)**
The farmer in Example 1.1.1 aims at solving the following canonical linear program: Minimise

$$f(x_1, x_2) = -100x_1 - 250x_2$$

subject to

$$\begin{aligned} x_1 + x_2 &\leq 40 \\ 40x_1 + 120x_2 &\leq 2400 \\ 6x_1 + 12x_2 &\leq 312 \\ x_1, x_2 &\geq 0 \end{aligned}$$

Define the linear functions

$$\begin{aligned} g_1(x_1, x_2) &:= x_1 + x_2, \\ g_2(x_1, x_2) &:= 40x_1 + 120x_2, \\ g_3(x_1, x_2) &:= 6x_1 + 12x_2. \end{aligned}$$

Let us investigate the first constraint $g_1(x_1, x_2) = x_1 + x_2 \leq 40$*, cf Figure 1.1. The equation* $g_1(x_1, x_2) = x_1 + x_2 = 40$ *defines a straight line in* $\mathbb{R}^2$ *– a so-called hyperplane*

$$H := \{(x_1, x_2)^\top \in \mathbb{R}^2 \mid x_1 + x_2 = 40\}.$$

The vector of coefficients $n_{g_1} = (1, 1)^\top$ *is the normal vector on the hyperplane* H *and it points in the direction of increasing values of the function* g_1*. I.e. moving a point on* H *into the direction of* n_{g_1} *leads to a point* $(x_1, x_2)^\top$ *with* $g_1(x_1, x_2) > 40$*. Moving a point on* H *into the opposite direction* $-n_{g_1}$ *leads to a point* $(x_1, x_2)^\top$ *with* $g_1(x_1, x_2) < 40$*. Hence,* H *separates* $\mathbb{R}^2$ *into two halfspaces defined by*

$$\begin{aligned} H^- &:= \{(x_1, x_2)^\top \in \mathbb{R}^2 \mid x_1 + x_2 \leq 40\}, \\ H^+ &:= \{(x_1, x_2)^\top \in \mathbb{R}^2 \mid x_1 + x_2 > 40\}. \end{aligned}$$

Obviously, points in H^+ *are not feasible, while points in* H^- *are feasible w.r.t. to the constraint* $x_1 + x_2 \leq 40$*.*

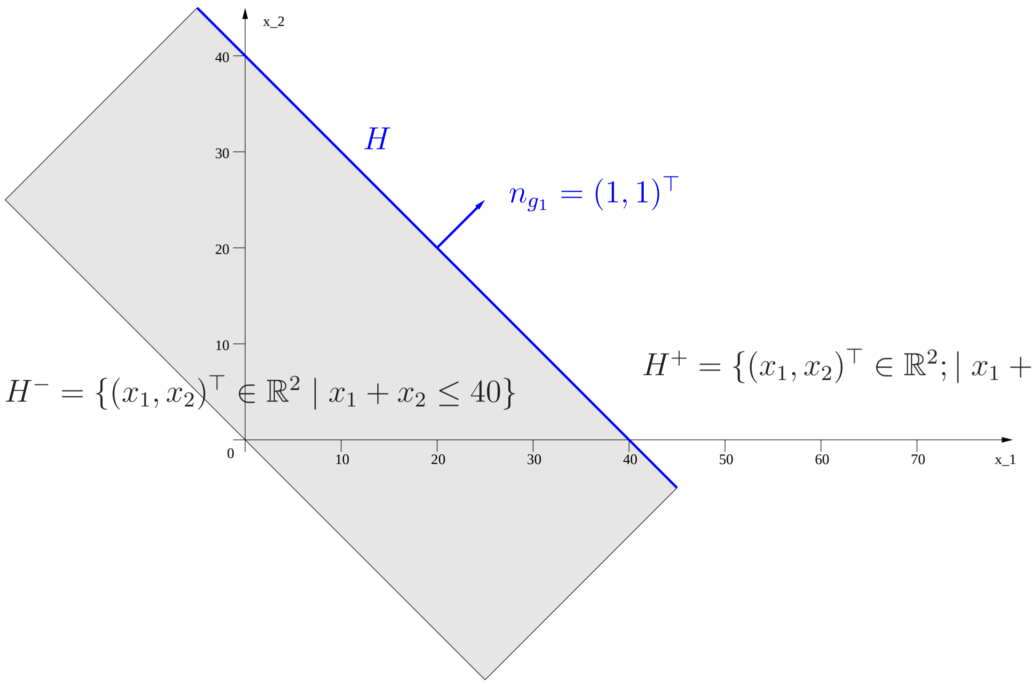


Figure 1.1: Geometry of the constraint $g_1(x_1, x_2) = x_1 + x_2 \leq 40$: normal vector $n_{g_1} = (1, 1)^\top$ (the length has been scaled), hyperplane H and halfspaces H^+ and H^-. All points in H^- are feasible w.r.t. to this constraint

The same discussion can be performed for the remaining constraints (don't forget the constraints $x_1 \geq 0$ *and* $x_2 \geq 0$*!). The feasible set is the dark area in Figure 1.2 given by the intersection of the respective halfspaces* H^- *of the constraints.*

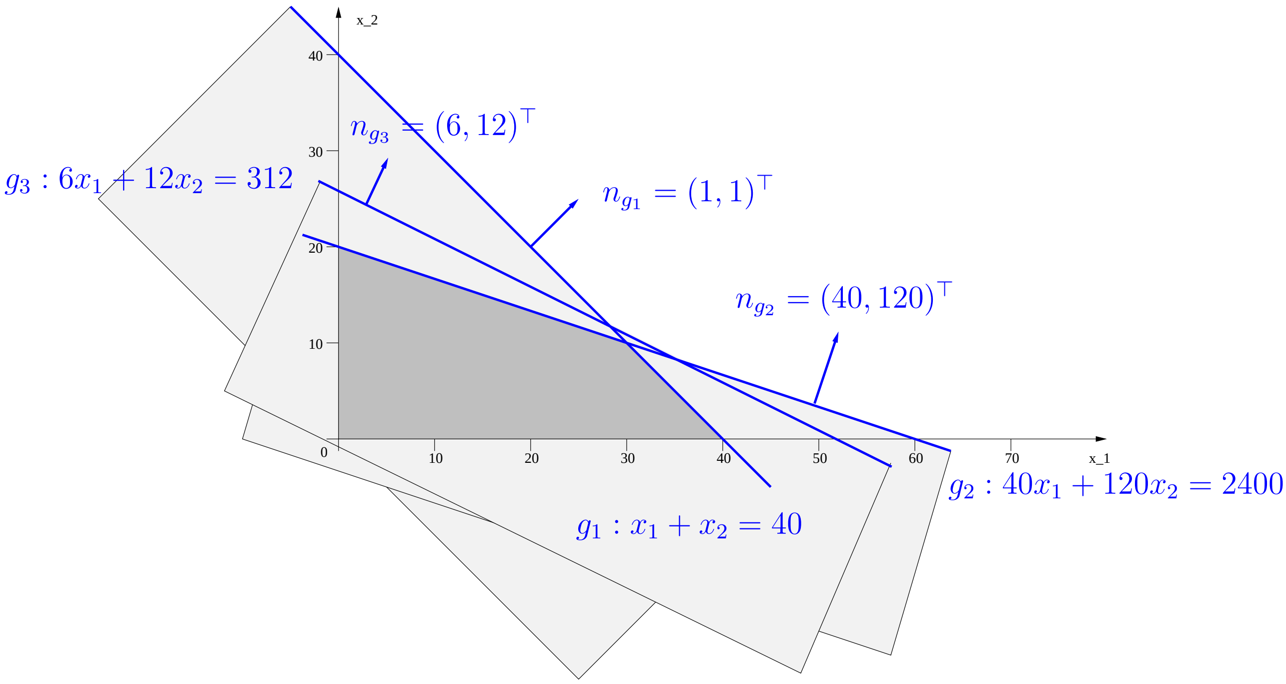


Figure 1.2: Feasible set of the linear program: Intersection of halfspaces for the constraints g_1, g_2, g_3 and $x_1 \geq 0$ and $x_2 \geq 0$ (the length of the normal vectors has been scaled).

We haven't considered the objective function so far. The red line in the Figure 1.3 corresponds to the line

$$f(x_1, x_2) = -100x_1 - 250x_2 = -3500,$$

i.e. for all points on this line the objective function assumes the value -3500. *Herein, the value* -3500 *is just an arbitrary guess of the optimal value. The green line corresponds to the line*

$$f(x_1, x_2) = -100x_1 - 250x_2 = -5500,$$

i.e. for all points on this line the objective function assumes the value -5500.
Obviously, the function values of f *increase in the direction of the normal vector* $n_f(x_1, x_2) = (-100, -250)^\top$, *which is nothing else than the gradient of* f. *As we intend to minimise* f, *we have to 'move' the 'objective function line' in the opposite direction* $-n_f$ *(direction of descent!) as long as it does not completely leave the feasible set (the intersection of this line and the feasible set has to be nonempty). The feasible points on this extremal 'objective function line' are then optimal. The green line is the optimal line as there would be no feasible points on it if we moved it any further in the direction* $-n_f$.
According to the figure we graphically determine the optimal solution to be $x_1 = 30$, $x_2 = 10$ *with objective function* $f(x_1, x_2) = -5500$. *The solution is attained in a vertex of the feasible set.*
In this example, the feasible vertices are given by the points $(0, 0)$, $(0, 20)$, $(30, 10)$, *and*

(40, 0).

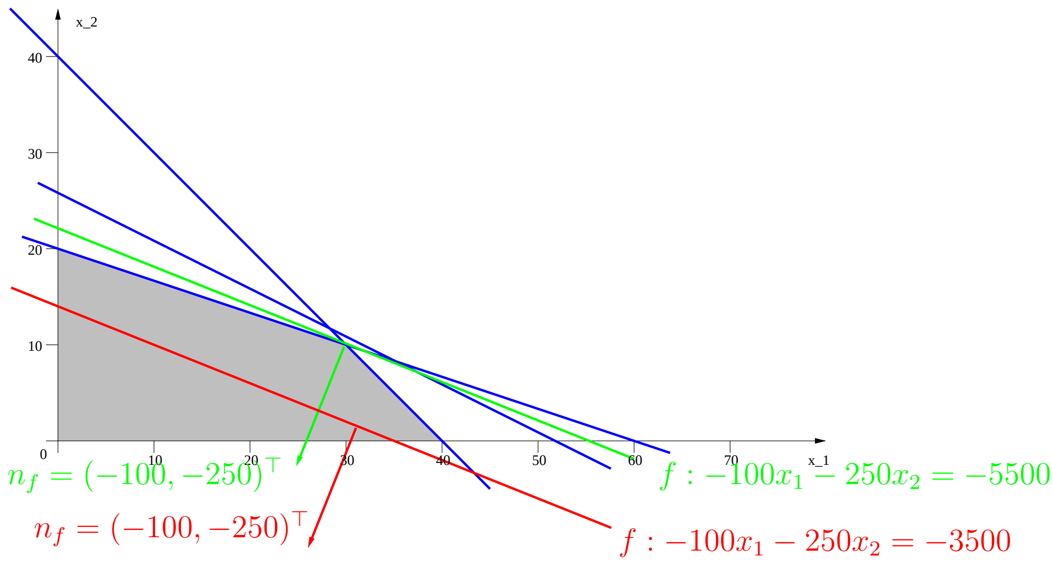


Figure 1.3: Solving the LP graphically: Move the objective function line in the opposite direction of n_f as long as the objective function line intersects with the feasible set (dark area). The green line is optimal. The optimal solution is $(x_1, x_2)^\top = (30, 10)^\top$.

In general, each row

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \leq b_i, \qquad i \in \{1, \ldots, m\},$$

of the constraint $Ax \leq b$ of the canonical linear program defines a hyperplane

$$H = \{x \in \mathbb{R}^n \mid a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n = b_i\},$$

with normal vector $(a_{i1}, \ldots, a_{in})^\top$, a positive halfspace

$$H^+ = \{x \in \mathbb{R}^n \mid a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \geq b_i\},$$

and a negative halfspace

$$H^- = \{x \in \mathbb{R}^n \mid a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \leq b_i\}.$$

Recall that the normal vector n is perpendicular to H and points into the positive halfspace H^+, cf. Figure*1.4.

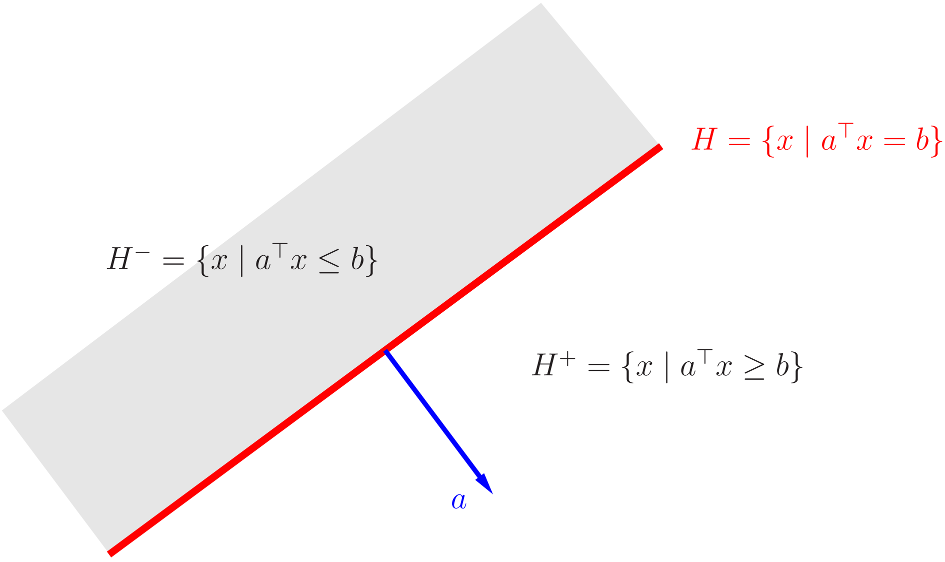


Figure 1.4: Geometry of the inequality $a^\top x \le b$: Normal vector a, hyperplane H, and halfspaces H^+ and H^-.

The feasible set M_C of the canonical linear program is given by the intersection of the respective negative halfspaces for the constraints $Ax \le b$ and $x \ge 0$. The intersection of a finite number of halfspaces is called polyedric set or polyeder. A bounded polyeder is called polytope. Figure 1.5 depicts the different constellations that may occur in linear programming.

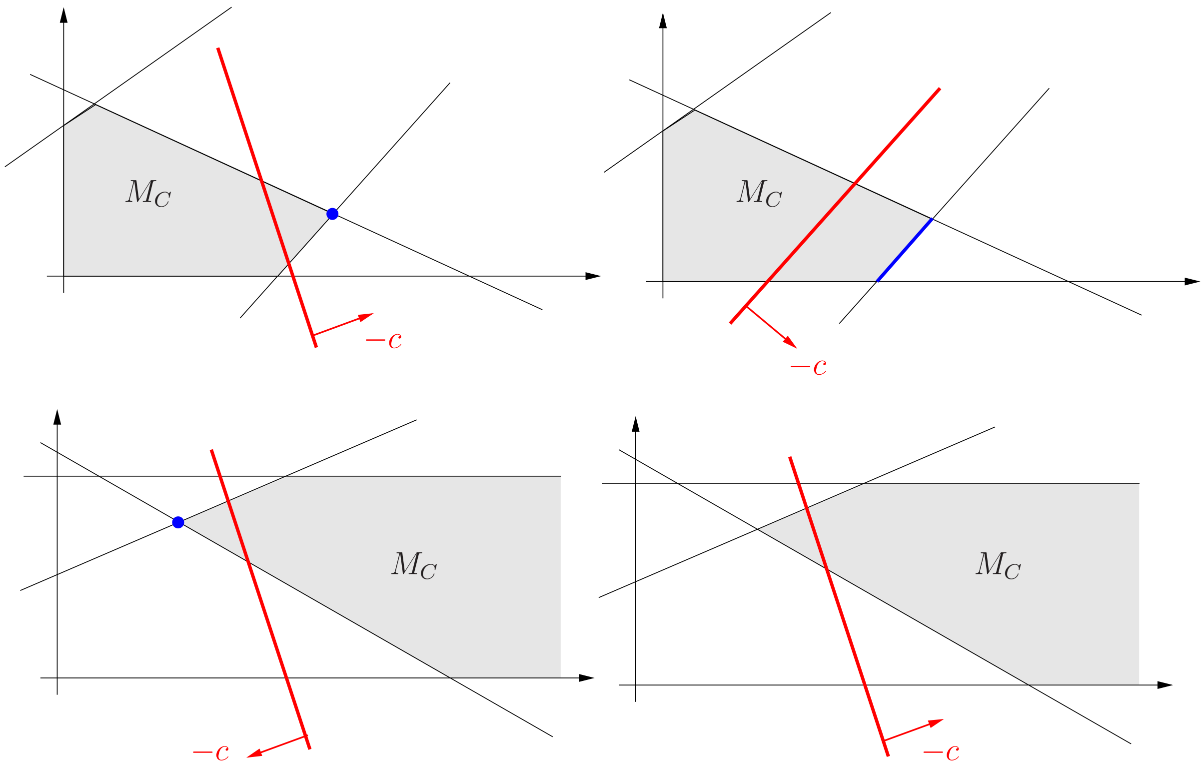


Figure 1.5: Different geometries of the feasible set (grey), objective function line (red), optimal solution (blue).

Summarising, the graphical solution of an LP works as follows:

(1) Sketch the feasible set.

(2) Make a guess $w \in \mathbb{R}$ of the optimal objective function value or alternatively take an arbitrary (feasible) point $\hat{x}$, introduce this point $\hat{x}$ into the objective function and compute $w = c^\top \hat{x}$. Plot the straight line given by $c^\top x = w$.

(3) Move the straight line in (2) in the direction $-c$ as long as the intersection of the line with the feasible set is non-empty.

(4) The most extreme line in (3) is optimal. All feasible points on this line are optimal. Their values can be found in the picture.

Observations:

- The feasible set M_C can be bounded (pictures on top) or unbounded (pictures at bottom).
- If the feasible set is bounded, there always exists an optimal solution by the Theorem of Weierstrass (M_C is compact and the objective function is continuous and thus assumes its minimum and maximum value on M_C).

- The optimal solution can be unique (pictures on left) or non-unique (top-right picture).
- The bottom-left figure shows that an optimal solution may exist even if the feasible set is unbounded.
- The bottom-right figure shows that the LP might not have an optimal solution.

The following observation is the main motivation for the simplex method. We will see that this observation is always true!

Main observation:

> If an optimal solution exists, there is always a 'vertex' of the feasible set among the optimal solutions.

1.4 The Fundamental Theorem of Linear Programming

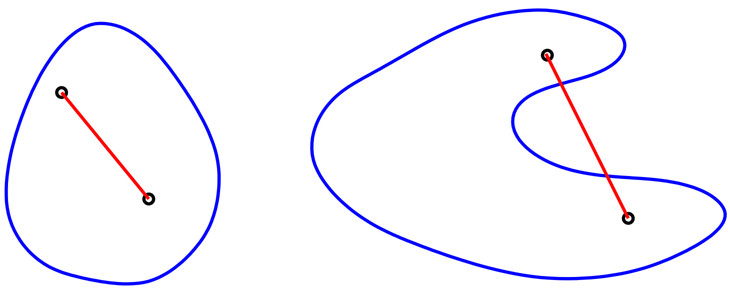
The main observation suggests that vertices play an outstanding role in linear programming. While it is easy to identify vertices in a picture, it is more difficult to characterise a vertex mathematically. We will define a vertex for convex sets.
We note

Theorem 1.4.1

The feasible set of a linear program is convex.

Herein, a set $M \in \mathbb{R}^n$ is called convex, if the entire line connecting two points in M belongs to M, i.e.

$$x, y \in M \quad \Rightarrow \quad \lambda x + (1-\lambda)y \in M \text{ for all } 0 \leq \lambda \leq 1.$$



convex and non-convex set

Proof: Without loss of generality the discussion is restricted to the standard LP. Let $x, y \in M_S$, $0 \leq \lambda \leq 1$, and $z = \lambda x + (1-\lambda)y$. We have to show that $z \in M_S$ holds. Since $\lambda \in [0,1]$ and $x, y \geq 0$ it holds $z \geq 0$. Moreover, it holds

$$Az = \lambda Ax + (1-\lambda)Ay = \lambda b + (1-\lambda)b = b.$$

Hence, $z \in M_S$. □

Now we define a vertex of a convex set.

Definition 1.4.2 (Vertex)

Let M be a convex set. $x \in M$ is called vertex of M, if the representation $x = \lambda x_1 + (1-\lambda)x_2$ with $0 < \lambda < 1$ and $x_1, x_2 \in M$ implies $x_1 = x_2 = x$.

Remark 1.4.3

The set $\{\lambda x_1 + (1-\lambda)x_2 \mid 0 < \lambda < 1\}$ is the straight line interconnecting the points x_1 and x_2 (x_1 and x_2 are not included). Hence, if x is a vertex it cannot be on this line unless $x = x_1 = x_2$.

According to the following theorem it suffices to visit only the vertices of the feasible set in order to find one (not every!) optimal solution. As any LP can be transformed into standard form, it suffices to restrict the discussion to standard LP's. Of course, the following result holds for any LP.

Theorem 1.4.4 (Fundamental Theorem of Linear Programming)

Let a standard LP 1.2.5 be given with $M_S \neq \emptyset$. Then:

(a) Either the objective function is unbounded from below on M_S or Problem 1.2.5 has an optimal solution and at least one vertex of M_S is among the optimal solutions.

(b) If M_S is bounded, then an optimal solution exists and $x \in M_S$ is optimal, if and only if x is a convex combination of optimal vertices.

Proof: The proof exploits the following facts about polyeders, which we don't proof here:

(i) If $M_S \neq \emptyset$, then at least one vertex exists.

(ii) The number of vertices of M_S is finite.

(iii) Let $x^1, \ldots, x^N$ with $N \in \mathbb{N}$ denote the vertices of M_S. For every $x \in M_S$ there exists a vector $d \in \mathbb{R}^n$ and scalars λ_i, $i = 1, \ldots, N$, such that

$$x = \sum_{i=1}^{N} \lambda_i x^i + d$$

and

$$\lambda_i \geq 0, \quad i = 1, \ldots, N, \quad \sum_{i=1}^{N} \lambda_i = 1, \quad d \geq 0, \quad Ad = 0.$$

(iv) If M_S is bounded, then every $x \in M_S$ can be expressed as

$$x = \sum_{i=1}^{N} \lambda_i x^i, \qquad \lambda_i \geq 0, \quad i = 1, \ldots, N, \quad \sum_{i=1}^{N} \lambda_i = 1.$$

With these facts the assertions can be proven as follows.

(a) Case 1: If in (iii) there exists a $d \in \mathbb{R}^n$ with $d \geq 0$, $Ad = 0$, and $c^\top d < 0$, then the objective function is unbounded from below as for an $x \in M_S$ the points $x + td$ with $t \geq 0$ are feasible as well because $A(x+td) = Ax+tAd = Ax = b$ and $x+td \geq x \geq 0$. But $c^\top(x + td) = c^\top x + tc^\top d \to -\infty$ for $t \to \infty$ and thus the objective function is unbounded from below on M_S.

Case 2: Let $c^\top d \geq 0$ hold for every $d \in \mathbb{R}^n$ with $d \geq 0$ and $Ad = 0$. According to (i), (ii), and (iii) every point $x \in M_S$ can be expressed as

$$x = \sum_{i=1}^{N} \lambda_i x^i + d$$

with suitable

$$\lambda_i \geq 0, \quad i = 1, \ldots, N, \quad \sum_{i=1}^{N} \lambda_i = 1, \quad d \geq 0, \quad Ad = 0.$$

Let x^0 be an optimal point. Then, x^0 can be expressed as

$$x^0 = \sum_{i=1}^{N} \lambda_i x^i + d, \qquad \lambda_i \geq 0, \quad \sum_{i=1}^{N} \lambda_i = 1, \quad Ad = 0, \quad d \geq 0.$$

Let x^j, $j \in \{1, \ldots, N\}$, denote the vertex with the minimal objective function value, i.e. it holds $c^\top x^j \leq c^\top x^i$ for all $i = 1, \ldots, N$. Then

$$c^\top x^0 = \sum_{i=1}^{N} \lambda_i c^\top x^i + c^\top d \geq \sum_{i=1}^{N} \lambda_i c^\top x^i \geq c^\top x^j \sum_{i=1}^{N} \lambda_i = c^\top x^j.$$

As x^0 was supposed to be optimal, we found that x^j is optimal as well which shows the assertion.

(b) Let M_S be bounded. Then the objective function is bounded from below on M_S according to the famous Theorem of Weierstrass. Hence, the LP has at least one solution according to (a). Let x^0 be an arbitrary optimal point. According to (iv) x^0 can be expressed as a convex combination of the vertices x^i, $i = 1, \ldots, N$, i.e.

$$x^0 = \sum_{i=1}^{N} \lambda_i x^i, \qquad \lambda_i \geq 0, \quad \sum_{i=1}^{N} \lambda_i = 1.$$

Let x^j, $j \in \{1, \ldots, N\}$, be an optimal vertex (which exists according to (a)), i.e.

$$c^\top x^0 = c^\top x^j = \min_{i=1,\ldots,N} c^\top x^i.$$

Then, using the expression for x^0 it follows

$$c^\top x^0 = \sum_{i=1}^{N} \lambda_i c^\top x^i = c^\top x^j = \min_{i=1,\ldots,N} c^\top x^i.$$

Now let x^k, $k \in \{1, \ldots, N\}$, be a non-optimal vertex with $c^\top x^k = c^\top x^0 + \varepsilon$, $\varepsilon > 0$. Then,

$$c^\top x^0 = \sum_{i=1}^{N} \lambda_i c^\top x^i = \sum_{i=1, i\neq k}^{N} \lambda_i c^\top x^i + \lambda_k c^\top x^k \geq c^\top x^0 \sum_{i=1, i\neq k}^{N} \lambda_i + \lambda_k c^\top x^0 + \lambda_k \varepsilon = c^\top x^0 + \lambda_k \varepsilon.$$

As $\varepsilon > 0$ this implies $\lambda_k = 0$ which shows the first part of the assertion.

If x^0 is a convex combination of optimal vertices, i.e.

$$x^0 = \sum_{i=1}^{N} \lambda_i x^i, \qquad \lambda_i \geq 0, \quad \sum_{i=1}^{N} \lambda_i = 1,$$

then due to the linearity of the objective function and the convexity of the feasible set it is easy to show that x^0 is optimal. This completes the proof.

□

1.5 Software for LP's

Matlab (www.mathworks.com) provides the command `linprog` for the solution of linear programs.
Scilab (www.scilab.org) is a powerful platform for scientific computing. It provides the command `linpro` for the solution of linear programs.
There are many commercial and non-commercial implementations of algorithms for linear programming on the web. For instance, the company **Lindo Systems Inc.** develops software for the solution of linear, integer, nonlinear and quadratic programs. On their webpage

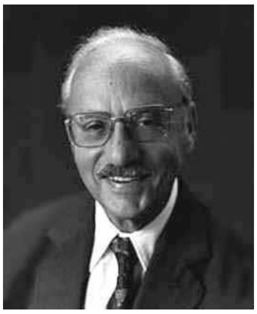
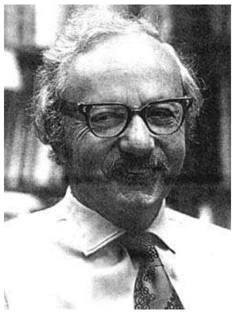
`http://www.lindo.com`

it is possible to download a free trial version of the software `LINDO`.
The program GeoGebra (www.geogebra.org) is a nice geometry program that can be used to visualise feasible sets.

Chapter 2

The Simplex Method

A real breakthrough in linear programming was the invention of the simplex method by G. B. Dantzig in 1947.

George Bernard Dantzig
Born: 8.11.1914 in Portland (Oregon)
Died: 13.5.2005 in Palo Alto (California)

The simplex method is one of the most important and popular algorithms for solving linear programs on a computer. The very basic idea of the simplex method is to move from a feasible vertex to a neighbouring feasible vertex and repeat this procedure until an optimal vertex is reached. According to Theorem 1.4.4 it is sufficient to consider only the vertices of the feasible set.
In order to construct the simplex algorithm, we need to answer the following questions:

- How can (feasible) vertices be computed?
- Given a feasible vertex, how can we compute a neighbouring feasible vertex?
- How to check for optimality?

In this chapter we will restrict the discussion to LP's in standard form 1.2.5, i.e.

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

Throughout the rest of the chapter we assume $\text{rank}(A) = m$. This assumption excludes linearly dependent constraints from the problem formulation.

Notation:

- The rows of A are denoted by

$$a_i^\top := (a_{i1}, \dots, a_{in}) \in \mathbb{R}^n, \qquad i = 1, \dots, m,$$

i.e.

$$A = \begin{pmatrix} \boxed{a_{11} \quad a_{12} \quad \cdots \quad a_{1n}} \\ \boxed{a_{21} \quad a_{22} \quad \cdots \quad a_{2n}} \\ \vdots \quad \vdots \quad \ddots \quad \vdots \\ \boxed{a_{m1} \quad a_{m2} \quad \cdots \quad a_{mn}} \end{pmatrix} = \begin{pmatrix} a_1^\top \\ a_2^\top \\ \vdots \\ a_m^\top \end{pmatrix}.$$

- The columns of A are denoted by

$$a^j := (a_{1j}, \dots, a_{mj})^\top \in \mathbb{R}^m, \qquad j = 1, \dots, n,$$

i.e.

$$A = \left(\begin{array}{|c|c|c|c|} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{array} \right) = \begin{pmatrix} a^1 & a^2 & \cdots & a^n \end{pmatrix}.$$

2.1 Vertices and Basic Solutions

The geometric definition of a vertex in Definition 1.4.2 is not useful if a vertex has to be computed explicitely within an algorithm. Hence, we are looking for an alternative characterisation of a vertex which allows us to actually compute a vertex. For this purpose the standard LP is particularly well suited.
Consider the feasible set of the standard LP 1.2.5:

$$M_S = \{x \in \mathbb{R}^n \mid Ax = b,\ x \geq 0\}.$$

Let $x = (x_1, \dots, x_n)^\top$ be an arbitrary point in M_S. Let

$$B := \{i \in \{1, \dots, n\} \mid x_i > 0\}$$

be the index set of positive components of x, and

$$N := \{1, \dots, n\} \setminus B = \{i \in \{1, \dots, n\} \mid x_i = 0\}$$

the index set of vanishing components of x. Because of $x \in M_S$ it holds

$$b = Ax = \sum_{j=1}^{n} a^j x_j = \sum_{j \in B} a^j x_j.$$

This is a linear equation for the components x_j, $j \in B$. This linear equation has a unique solution, if the column vectors a^j, $j \in B$, are linearly independent. In this case, we call x a feasible basic solution.

Definition 2.1.1 **(feasible basic solution)**

Let $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, and $M_S = \{x \in \mathbb{R}^n \mid Ax = b,\ x \geq 0\}$.
$x \in M_S$ is called feasible basic solution (for the standard LP), if the column vectors $\{a^j \mid j \in B\}$ are linearly independent, where $B = \{i \in \{1,\dots,n\} \mid x_i > 0\}$.

The following theorem states that a feasible basic solution is a vertex of the feasible set and vice versa.

Theorem 2.1.2

$x \in M_S$ is a vertex of M_S if and only if x is a feasible basic solution.

Proof: Recall that the feasible set M_S is convex, see Theorem 1.4.1.

'$\Rightarrow$' Let x be a vertex. Without loss of generality let $x = (x_1,\dots,x_r,0,\dots,0)^\top \in \mathbb{R}^n$ with $x_i > 0$ for $i = 1,\dots,r$.

We assume that a^j, $j = 1,\dots,r$, are linearly dependent. Then there exist α_j, $j = 1,\dots,r$, not all of them are zero with

$$\sum_{j=1}^{r} \alpha_j a^j = 0.$$

Define

$$\begin{aligned} y_+ &= (x_1 + \varepsilon\alpha_1,\dots,x_r + \varepsilon\alpha_r,0,\dots,0)^\top \\ y_- &= (x_1 - \varepsilon\alpha_1,\dots,x_r - \varepsilon\alpha_r,0,\dots,0)^\top \end{aligned}$$

with

$$0 < \varepsilon < \min\left\{ \frac{|x_j|}{|\alpha_j|} \;\middle|\; \alpha_j \neq 0,\ 1 \leq j \leq r \right\}.$$

According to this choice of ε it follows $y_+, y_- \geq 0$. Moreover, y_+ and y_- are feasible because

$$Ay_\pm = \sum_{j=1}^{r}(x_j \pm \varepsilon\alpha_j)a^j = \sum_{j=1}^{r} x_j a^j \pm \varepsilon \underbrace{\sum_{j=1}^{r}\alpha_j a^j}_{=0,\ \text{lin. dep.}} = b.$$

Hence, $y_+, y_- \in M_S$, but $(y_+ + y_-)/2 = x$. This contradicts the assumption that x is a vertex.

'$\Leftarrow$' Without loss of generality let $x = (x_1,\dots,x_r,0,\dots,0)^\top \in M_S$ with $x_j > 0$ and a^j linearly independent for $j = 1,\dots,r$. Let $y, z \in M_S$ and $x = \lambda y + (1-\lambda)z$, $0 < \lambda < 1$. Then,

(i)

$$x_j > 0 \quad (1 \leq j \leq r) \quad \Rightarrow \quad y_j > 0 \text{ or } z_j > 0 \text{ for all } 1 \leq j \leq r.$$

(ii)

$$x_j = 0 \quad (j = r+1,\ldots,n) \quad \Rightarrow \quad y_j = z_j = 0 \text{ for } r+1 \le j \le n.$$

$b = Ay = Az$ implies $0 = A(y-z) = \sum_{j=1}^{r}(y_j - z_j)a^j$. Due to the linear independence of a^j it follows $y_j - z_j = 0$ for $j = 1,\ldots,r$. Since $y_j = z_j = 0$ for $j = r+1,\ldots,n$, it holds $y = z$. Hence, x is a vertex.

□

2.2 The (primal) Simplex Method

Theorem 2.1.2 states that a vertex can be characterised by linearly independent columns of A. According to Theorem 1.4.4 it is sufficient to compute only the vertices (feasible basic solutions) of the feasible set in order to obtain at least one optimal solution – provided such a solution exists at all. This is the basic idea of the simplex method.

Notation:
Let $B \subseteq \{1,\ldots,n\}$ be an index set.

- Let $x = (x_1,\ldots,x_n)^\top$ be a vector. Then, x_B is defined to be the vector with components x_i, $i \in B$.
- Let A be a $m \times n$-matrix with columns a^j, $j = 1,\ldots,n$. Then, A_B is defined to be the matrix with columns a^j, $j \in B$.

Example:

$$x = \begin{pmatrix} -1 \\ 10 \\ 3 \\ -4 \end{pmatrix}, \quad A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{pmatrix}, \quad B = \{2,4\} \quad \Rightarrow \quad x_B = \begin{pmatrix} 10 \\ -4 \end{pmatrix}, \quad A_B = \begin{pmatrix} 2 & 4 \\ 6 & 8 \end{pmatrix}.$$

The equivalence theorem 2.1.2 is of fundamental importance for the simplex method, because we can try to find vertices by solving linear equations as follows:

Algorithm 2.2.1 (Computation of a vertex)

(1) *Choose m linearly independent columns a^j, $j \in B$, $B \subseteq \{1,\ldots,n\}$, of A and set $N = \{1,\ldots,n\} \setminus B$.*

(2) *Set $x_N = 0$ and solve the linear equation*

$$A_B x_B = b.$$

(3) If $x_B \geq 0$, x is a feasible basic solution and thus a vertex. STOP. If there exists an index $i \in B$ with $x_i < 0$, then x is infeasible. Go to (1) and repeat the procedure with a different choice of linearly independent columns.

Recall $\text{rank}(A) = m$, which guarantees the existence of m linearly independent columns of A in step (1) of the algorithm.

A naive approach to solve a linear program would be to compute all vertices of the feasible set by the above algorithm. As there are at most $\binom{n}{m}$ ways to choose m linearly independent columns from a total of n columns there exist at most

$$\binom{n}{m} = \frac{n!}{m!(n-m)!}$$

vertices (resp. feasible basic solutions). Unfortunately, this is a potentially large number, especially if m and n are large (1 million, say). Hence, it is not efficient to compute every vertex.

Caution: Computing all vertices and choosing that one with the smallest objective function value may not be sufficient to solve the problem, because it may happen that the feasible set and the objective function is unbounded! In this case, no solution exists.

Example 2.2.2

Consider the constraints $Ax = b$, $x \geq 0$, with

$$A = \begin{pmatrix} 2 & 3 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}, \qquad b = \begin{pmatrix} 6 \\ 2 \end{pmatrix}.$$

There are at most $\binom{4}{2} = 6$ possible combinations of two columns of A:

$$\begin{aligned}
B_1 = \{1,2\} &\Rightarrow x_{B_1} = A_{B_1}^{-1}b = \begin{pmatrix} 2 \\ \frac{2}{3} \end{pmatrix}, \quad x^1 = \left(2, \frac{2}{3}, 0, 0\right)^\top, \\
B_2 = \{1,3\} &\Rightarrow x_{B_2} = A_{B_2}^{-1}b = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \quad x^2 = (2,0,2,0)^\top, \\
B_3 = \{1,4\} &\Rightarrow x_{B_3} = A_{B_3}^{-1}b = \begin{pmatrix} 3 \\ -1 \end{pmatrix}, \quad x^3 = (3,0,0,-1)^\top, \\
B_4 = \{2,3\} &\Rightarrow A_B \text{ is singular}, \\
B_5 = \{2,4\} &\Rightarrow x_{B_5} = A_{B_5}^{-1}b = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \quad x^5 = (0,2,0,2)^\top, \\
B_6 = \{3,4\} &\Rightarrow x_{B_6} = A_{B_6}^{-1}b = \begin{pmatrix} 6 \\ 2 \end{pmatrix}, \quad x^6 = (0,0,6,2)^\top.
\end{aligned}$$

Herein, those components x_j with $j \notin B$ are set to 0. x^3 is not feasible. Hence, the vertices are given by x^1, x^2, x^5, x^6.

Remark 2.2.3

It is possible to define basic solutions which are not feasible. Let $B \subseteq \{1, \ldots, n\}$ be an index set with $|B| = m$, $N = \{1, \ldots, n\} \setminus B$, and let the columns a^i, $i \in B$, be linearly independent. Then, x is called a basic solution if

$$A_B x_B = b, \qquad x_N = 0.$$

Notice that such a $x_B = A_B^{-1} b$ does not necessarily satisfy the sign condition $x_B \geq 0$, i.e. the above x may be infeasible. Those infeasible basic solutions are not of interest for the linear program.

We need some more definitions.

Definition 2.2.4

- *(Basis)*
  Let rank$(A) = m$ and let x be a feasible basic solution of the standard LP. Every system $\{a^j \mid j \in B\}$ of m linearly independent columns of A, which includes those columns a^j with $x_j > 0$, is called basis of x.

- *((Non-)basis index set, (non-)basis matrix, (non-)basic variable)*
  Let $\{a^j \mid j \in B\}$ be a basis of x. The index set B is called basis index set, the index set $N := \{1, \ldots, n\} \setminus B$ is called non-basis index set, the matrix $A_B := (a^j)_{j \in B}$ is called basis matrix, the matrix $A_N := (a^j)_{j \in N}$ is called non-basis matrix, the vector $x_B := (x_j)_{j \in B}$ is called basic variable and the vector $x_N := (x_j)_{j \in N}$ is called non-basic variable.

Let's consider an example.

Example 2.2.5

Consider the inequality constraints

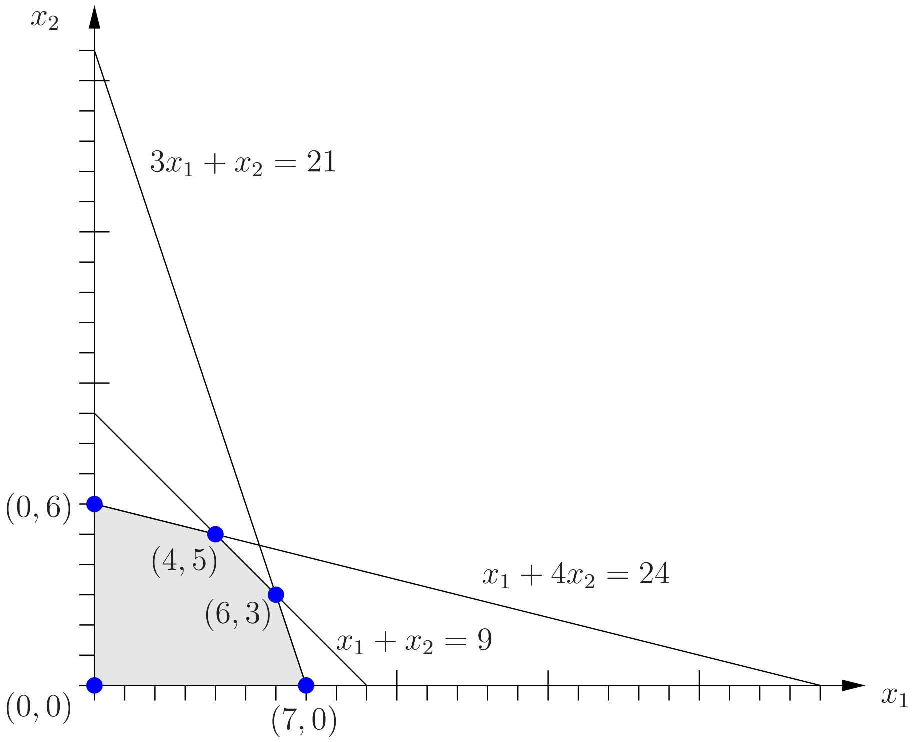
$$\begin{aligned} x_1 + 4x_2 &\leq 24, \\ 3x_1 + x_2 &\leq 21, \\ x_1 + x_2 &\leq 9, \qquad x_1 \geq 0,\ x_2 \geq 0. \end{aligned}$$

Introduction of slack variables $x_3 \geq 0$, $x_4 \geq 0$, $x_5 \geq 0$ leads to standard constraints

$Ax = b$, $x \geq 0$, with

$$A = \begin{pmatrix} 1 & 4 & 1 & 0 & 0 \\ 3 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}, \qquad b = \begin{pmatrix} 24 \\ 21 \\ 9 \end{pmatrix}, \qquad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix}.$$

The projection of the feasible set $M_S = \{x \in \mathbb{R}^5 \mid Ax = b,\ x \geq 0\}$ into the (x_1, x_2)-plane looks as follows:



(i) Consider $x = (6,3,6,0,0)^\top \in M_S$. The first three components are positive and the corresponding columns of A, i.e. $\begin{pmatrix} 1 \\ 3 \\ 1 \end{pmatrix}$, $\begin{pmatrix} 4 \\ 1 \\ 1 \end{pmatrix}$ and $\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$, are linearly independent. x is actually a feasible basic solution and according to Theorem 2.1.2 a vertex. We obtain the basis

$$\left\{ \begin{pmatrix} 1 \\ 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 4 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \right\}$$

with basis index set $B = \{1,2,3\}$, non-basis index set $N = \{4,5\}$, basic variable $x_B = (6,3,6)^\top$, non-basic variable $x_N = (0,0)^\top$ and basis matrix resp. non-basis *matrix*

$$A_B := \begin{pmatrix} 1 & 4 & 1 \\ 3 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}, \qquad A_N := \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

(ii) *Exercise: Consider as in (i) the remaining vertices which are given by* $(x_1, x_2) = (0, 0)$, $(x_1, x_2) = (0, 6)$, $(x_1, x_2) = (7, 0)$, *and* $(x_1, x_2) = (4, 5)$.

With these definition, we are ready to formulate a draft version of the simplex method :

(0) Phase 0:
Transform the linear program into standard form 1.2.5, if necessary at all.

(1) Phase 1:
Determine a feasible basic solution (feasible vertex) x with basis index set B, non-basis index set N, basic variable $x_B \geq 0$ and non-basic variable $x_N = 0$.

(2) Phase 2:
Compute a neighbouring feasible basic solution x^+ with basis index set B^+ and non-basis index set N^+ until either an optimum is computed or it can be decided that no such solution exists.

2.2.1 Computing a neighbouring feasible basic solution

In this section we will discuss how a neighbouring feasible basic solution x^+ can be computed in phase two of the simplex algorithm.
In the sequel, we assume that a feasible basic solution x with basis index set B and non-basis index set N is given. We will construct a new feasible basic solution x^+ with index sets

$$\begin{aligned} B^+ &= (B \setminus \{p\}) \cup \{q\}, \\ N^+ &= (N \setminus \{q\}) \cup \{p\} \end{aligned}$$

by interchanging suitably chosen indices $p \in B$ and $q \in N$. This procedure is called basis change.
The indices p and q are not chosen arbitrarily but in such a way that the following requirements are satisfied:

(i) Feasibility:
x^+ has to remain feasible, i.e.

$$Ax = b, \ x \geq 0 \qquad \Rightarrow \qquad Ax^+ = b, \ x^+ \geq 0.$$

(ii) Descent property:
The objective function value decreases monotonically, i.e.

$$c^\top x^+ \leq c^\top x.$$

Let a basis with basis index set B be given and let x be a feasible basic solution. Then,

- $A_B = (a^i)_{i\in B}$ is non-singular.
- $x_N = 0$.

Hence, the constraint $Ax = b$ can be solved w.r.t. the basic variable x_B according to

$$Ax = A_B x_B + A_N x_N = b \qquad \Rightarrow \qquad x_B = \underbrace{A_B^{-1}b}_{=\beta} - \underbrace{A_B^{-1}A_N}_{=\Gamma} x_N =: \beta - \Gamma x_N. \tag{2.1}$$

Introducing this expression into the objective function yields the expression

$$c^\top x = c_B^\top x_B + c_N^\top x_N = \underbrace{c_B^\top \beta}_{=d} - \underbrace{(c_B^\top \Gamma - c_N^\top)}_{=\zeta^\top} x_N =: d - \zeta^\top x_N. \tag{2.2}$$

A feasible basic solution x satisfies $x_N = 0$ in (2.1) and (2.2) and thus,

$$x_B = \beta \geq 0, \quad x_N = 0, \quad c^\top x = d. \tag{2.3}$$

Herein, we used the notation: $\beta = (\beta_i)_{i\in B}$, $\zeta = (\zeta_j)_{j\in N}$, $\Gamma = (\gamma_{ij})_{i\in B, j\in N}$.
Now we intend to change the basis in order to get to a neighbouring basis. Therefore, we consider the ray

$$z(t) = \begin{pmatrix} z_1(t) \\ \vdots \\ z_n(t) \end{pmatrix} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} + t \begin{pmatrix} s_1 \\ \vdots \\ s_n \end{pmatrix} = x + ts, \qquad t \geq 0, \tag{2.4}$$

emanating from the current basic solution x in direction s with step length $t \geq 0$, cf. Figure 2.1.

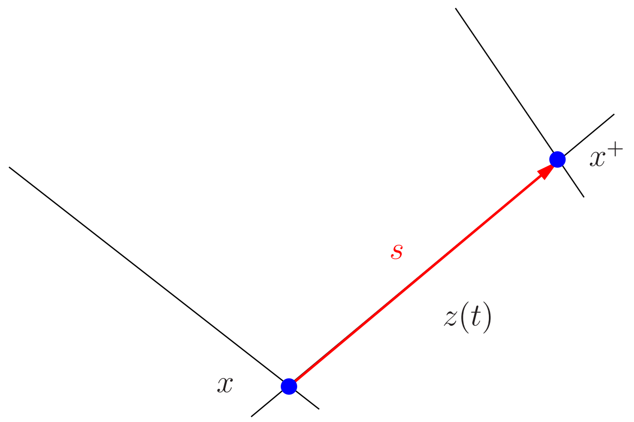


Figure 2.1: Idea of the basis change in the simplex method. Find a search direction s such that the objective function decreases monotonically along this direction.

We choose the search direction s in such a way that only the non-basic variable x_q with suitable index $q \in N$ will be changed while the remaining components $x_j = 0$, $j \in N$, $j \neq q$, remain unchanged. For a suitable index $q \in N$ we define

$$s_q := 1, \qquad s_j := 0 \text{ for } j \in N,\ j \neq q, \tag{2.5}$$

and thus,

$$z_q(t) = t \geq 0, \qquad z_j(t) = 0 \text{ for } j \in N,\ j \neq q. \tag{2.6}$$

Of course, the point $z(t)$ ought to be feasible, i.e. as in (2.1) the following condition has to hold:

$$b = Az(t) = \underbrace{Ax}_{=b} + tAs \quad \Rightarrow \quad 0 = As = A_B s_B + A_N s_N.$$

Solving this equation leads to

$$s_B = -A_B^{-1} A_N s_N = -\Gamma s_N. \tag{2.7}$$

Thus, the search direction s is completely defined by (2.5) and (2.7). Along the ray $z(t)$ the objective function value computes to

$$\begin{aligned} c^\top z(t) &\overset{(2.4)}{=} c^\top x + t c^\top s \\ &\overset{(2.3)}{=} d + t c_B^\top s_B + t c_N^\top s_N \\ &\overset{(2.7)}{=} d - t \left(c_B^\top \Gamma - c_N^\top \right) s_N \\ &= d - t \zeta^\top s_N \\ &\overset{(2.5)}{=} d - t \zeta_q. \end{aligned} \tag{2.8}$$

This representation immediately reveals that the objective function value decreases along s for $t \geq 0$, if $\zeta_q > 0$ holds.

If, on the other hand, $\zeta_j \leq 0$ holds for all $j \in N$, then a descent along s in the objective function is impossible. It remains to be investigated whether there are other points, which are not on the ray, but possibly lead to a smaller objective function value. This is not the case as for an arbitrary $\hat{x} \in M_S$ it holds $\hat{x} \geq 0$ and $A_B \hat{x}_B + A_N \hat{x}_N = b$ resp. $\hat{x}_B = \beta - \Gamma \hat{x}_N$. With $\zeta_j \leq 0$ and $\hat{x}_N \geq 0$ it follows

$$c^\top \hat{x} = c_B^\top \hat{x}_B + c_N^\top \hat{x}_N = c_B^\top \beta - (c_B^\top \Gamma - c_N^\top) \hat{x}_N = d - \zeta^\top \hat{x}_N \geq d = c^\top x.$$

Hence, if $\zeta_j \leq 0$ holds for all $j \in N$, then the current basic solution x is optimal! We summarise:

Choice of the pivot column q:
In order to fulfil the descent property in (ii), the index $q \in N$ has to be chosen such that $\zeta_q > 0$. If $\zeta_j \leq 0$ for all $j \in N$, then the current basic solution x is optimal.

Now we aim at satisfying the feasibility constraint in (i). The following has to hold:

$$z_B(t) = x_B + t s_B \overset{(2.7)}{=} \beta - t\Gamma s_N \geq 0,$$

respectively

$$\begin{aligned} z_i(t) &= \beta_i - t \sum_{j \in N} \gamma_{ij} s_j \\ &= \beta_i - t\gamma_{iq} \underbrace{s_q}_{=1} - t \sum_{j \in N, j \neq q} \gamma_{ij} \underbrace{s_j}_{=0} \\ &\overset{(2.5)}{=} \beta_i - \gamma_{iq} t \geq 0, \qquad i \in B. \end{aligned}$$

The conditions $\beta_i - \gamma_{iq} t \geq 0$, $i \in B$, restrict the step size $t \geq 0$. Two cases may occur:

(a) Case 1: It holds $\gamma_{iq} \leq 0$ for all $i \in B$.
Due to $\beta_i \geq 0$ it holds $z_i(t) = \beta_i - \gamma_{iq} t \geq 0$ for every $t \geq 0$ and every $i \in B$. Hence, $z(t)$ is feasible for every $t \geq 0$.

If in addition $\zeta_q > 0$ holds, then the objective function is not bounded from below for $t \to \infty$ according to (2.8). Thus, the linear program does not have a solution!

Unsolvability of LP
If for some $q \in N$ it holds $\zeta_q > 0$ and $\gamma_{iq} \leq 0$ for every $i \in B$, then the LP does not have a solution. The objective function is unbounded from below.

(b) Case 2: It holds $\gamma_{iq} > 0$ for at least one $i \in B$.
$\beta_i - \gamma_{iq} t \geq 0$ implies $t \leq \frac{\beta_i}{\gamma_{iq}}$. This postulation restricts the step length t.

Choice of pivot row p:
The feasibility in (i) will be satisfied by choosing an index $p \in B$ with

$$t_{min} := \frac{\beta_p}{\gamma_{pq}} := \min \left\{ \frac{\beta_i}{\gamma_{iq}} \;\middle|\; \gamma_{iq} > 0,\ i \in B \right\}.$$

It holds

$$\begin{aligned}
z_p(t_{min}) &= \beta_p - \gamma_{pq} t_{min} = \beta_p - \gamma_{pq}\frac{\beta_p}{\gamma_{pq}} = 0, \\
z_i(t_{min}) &= \underbrace{\beta_i}_{\geq 0} - \underbrace{\gamma_{iq}}_{\leq 0} \underbrace{t_{min}}_{\geq 0} \geq 0, \qquad \text{for } i \text{ with } \gamma_{iq} \leq 0, \\
z_i(t_{min}) &= \beta_i - \gamma_{iq} \underbrace{t_{min}}_{\leq \frac{\beta_i}{\gamma_{iq}}} \geq \beta_i - \gamma_{iq}\frac{\beta_i}{\gamma_{iq}} = 0 \qquad \text{for } i \text{ with } \gamma_{iq} > 0.
\end{aligned}$$

Hence, the point $x^+ := z(t_{min})$ is feasible and satisfies $x_p = 0$. Hence, x_p leaves the basic variables and enters the non-basic variables.

The following theorem states that x^+ is actually a feasible basic solution.

Theorem 2.2.6

Let x be a feasible basic solution with basis index set B. Let a pivot column $q \in N$ with $\zeta_q > 0$ and a pivot row $p \in B$ with $\gamma_{pq} > 0$ exist. Then, $x^+ = z(t_{min})$ is a feasible basic solution and $c^\top x^+ \leq c^\top x$. In particular, $\{a^j \mid j \in B^+\}$ with $B^+ = (B \setminus \{p\}) \cup \{q\}$ is a basis and A_{B^+} is non-singular.

Proof: By construction x^+ is feasible. It remains to show that $\{a^j \mid j \in B^+\}$ with $B^+ = (B \setminus \{p\}) \cup \{q\}$ is a basis. We note that $x_j = 0$ for all $j \notin B^+$.
The definition of $\Gamma = A_B^{-1} A_N$ implies $A_N = A_B \Gamma$. The column q of this matrix equation reads as

$$a^q = \sum_{i \in B} a^i \gamma_{iq} = a^p \gamma_{pq} + \sum_{i \in B, i \neq p} a^i \gamma_{iq}, \tag{2.9}$$

where γ_{iq} are the components of column q of Γ and $\gamma_{pq} \neq 0$ according to the assumption of this theorem.
We will now show that the vectors a^i, $i \in B$, $i \neq p$, and a^q are linearly independent. Therefore, we consider the equation

$$\alpha a^q + \sum_{i \in B, i \neq p} \alpha_i a^i = 0 \tag{2.10}$$

with coefficients $\alpha \in \mathbb{R}$ and $\alpha_i \in \mathbb{R}$, $i \in B$, $i \neq p$. We introduce a^q from (2.9) into (2.10) and obtain

$$\alpha \gamma_{pq} a^p + \sum_{i \in B, i \neq p} (\alpha_i + \alpha \gamma_{iq}) a^i = 0.$$

As $\{a^i \mid i \in B\}$ is a basis, we immediately obtain that all coefficients vanish, i.e.

$$\alpha\gamma_{pq} = 0, \qquad \alpha_i + \alpha\gamma_{iq} = 0, \qquad i \in B, i \neq p.$$

Since $\gamma_{pq} \neq 0$ it follows $\alpha = 0$ and this in turn implies $\alpha_i = 0$ for all $i \in B$, $i \neq p$. Hence, all coefficients vanish. This shows the linear independence of the set $\{a^i \mid i \in B^+\}$. □

2.2.2 The Algorithm

We summarise our findings in an algorithm:

Algorithm 2.2.7 **((Primal) Simplex Method)**

(0) *Phase 0:*

Transform the linear program into standard form 1.2.5, if necessary at all.

(1) *Phase 1:*

Determine a feasible basic solution (feasible vertex) x for the standard LP 1.2.5 with basis index set B, non-basis index set N, basis matrix A_B, non-basis matrix A_N, basic variable $x_B \geq 0$ and non-basic variable $x_N = 0$.

If no feasible solution exists, STOP. The problem is infeasible.

(2) *Phase 2:*

(i) *Compute $\Gamma = (\gamma_{ij})_{i\in B, j\in N}$, $\beta = (\beta_i)_{i\in B}$, and $\zeta = (\zeta_j)_{j\in N}$ according to*

$$\Gamma = A_B^{-1}A_N, \qquad \beta = A_B^{-1}b, \qquad \zeta^\top = c_B^\top\Gamma - c_N^\top.$$

(ii) *Check for optimality:*

If $\zeta_j \leq 0$ for every $j \in N$, then STOP. The current feasible basic solution $x_B = \beta$, $x_N = 0$ is optimal. The objective function value is $d = c_B^\top\beta$.

(iii) *Check for unboundedness:*

If there exists an index q with $\zeta_q > 0$ and $\gamma_{iq} \leq 0$ for every $i \in B$, then the linear program does not have a solution and the objective function is unbounded from below. STOP.

(iv) *Determine pivot element:*

Choose an index q with $\zeta_q > 0$. q defines the pivot column. Choose an index p with

$$\frac{\beta_p}{\gamma_{pq}} = \min\left\{ \frac{\beta_i}{\gamma_{iq}} \;\middle|\; \gamma_{iq} > 0,\ i \in B \right\}.$$

p defines the pivot row.

(v) *Perform basis change:*

Set $B := (B \setminus \{p\}) \cup \{q\}$ and $N := (N \setminus \{q\}) \cup \{p\}$.

(vi) *Go to (i).*

Remark 2.2.8

It is very important to recognise, that the indexing of the elements of the matrix Γ and the vectors β and ζ in (i) depends on the current entries and the orders in the index sets B and N. Notice that B and N are altered in each step (v) of the algorithm. So, Γ, β, and ζ are altered as well in each iteration. For instance, if $B = \{2,4,5\}$ and $N = \{1,3\}$, the entries of the matrix Γ and the vectors β and ζ are indexed as follows:

$$\Gamma = \begin{pmatrix} \gamma_{21} & \gamma_{23} \\ \gamma_{41} & \gamma_{43} \\ \gamma_{51} & \gamma_{53} \end{pmatrix}, \qquad \beta = \begin{pmatrix} \beta_2 \\ \beta_4 \\ \beta_5 \end{pmatrix}, \qquad \zeta = \begin{pmatrix} \zeta_1 \\ \zeta_3 \end{pmatrix}.$$

More generally, if $B = \{i_1, \ldots, i_m\}$, $i_k \in \{1, \ldots, n\}$, and $N = \{j_1, \ldots, j_{n-m}\}$, $j_k \in \{1, \ldots, n\} \setminus B$, then

$$\Gamma = \begin{pmatrix} \gamma_{i_1 j_1} & \gamma_{i_1 j_2} & \cdots & \gamma_{i_1 j_{n-m}} \\ \gamma_{i_2 j_1} & \gamma_{i_2 j_2} & \cdots & \gamma_{i_2 j_{n-m}} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{i_m j_1} & \gamma_{i_m j_2} & \cdots & \gamma_{i_m j_{n-m}} \end{pmatrix}, \qquad \beta = \begin{pmatrix} \beta_{i_1} \\ \beta_{i_2} \\ \vdots \\ \beta_{i_m} \end{pmatrix}, \qquad \zeta = \begin{pmatrix} \zeta_{j_1} \\ \zeta_{j_2} \\ \vdots \\ \zeta_{j_{n-m}} \end{pmatrix}.$$

Example 2.2.9 (compare Example 2.2.5)

Consider the standard LP

$$\text{Minimise} \quad c^\top x \quad \text{subject to} \quad Ax = b, \ x \geq 0$$

with the data (x_3, x_4, x_5 are slack variables):

$$c = \begin{pmatrix} -2 \\ -5 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \qquad A = \begin{pmatrix} 1 & 4 & 1 & 0 & 0 \\ 3 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}, \qquad b = \begin{pmatrix} 24 \\ 21 \\ 9 \end{pmatrix}, \qquad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix}.$$

Phase 1:

A feasible basic solution is given by the basis index set $B = \{3,4,5\}$, because the columns 3,4,5 of A are obviously linearly independent and

$$x_B = \begin{pmatrix} x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 24 \\ 21 \\ 9 \end{pmatrix} > 0 \qquad \text{and} \qquad x_N = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

is feasible, i.e. $x \geq 0$ *and* $Ax = b$.

Phase 2:

Iteration 0:

With $B = \{3,4,5\}$ *and* $N = \{1,2\}$ *we find*

$$A_B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \; A_N = \begin{pmatrix} 1 & 4 \\ 3 & 1 \\ 1 & 1 \end{pmatrix}, \; c_B = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \; c_N = \begin{pmatrix} -2 \\ -5 \end{pmatrix}.$$

and

$$\begin{pmatrix} \gamma_{31} & \gamma_{32} \\ \gamma_{41} & \gamma_{42} \\ \gamma_{51} & \gamma_{52} \end{pmatrix} = \begin{pmatrix} 1 & 4 \\ 3 & 1 \\ 1 & 1 \end{pmatrix}, \; \begin{pmatrix} \beta_3 \\ \beta_4 \\ \beta_5 \end{pmatrix} = \begin{pmatrix} 24 \\ 21 \\ 9 \end{pmatrix}, \; \begin{pmatrix} \zeta_1 \\ \zeta_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 5 \end{pmatrix}$$

The checks for optimality and unboundedness fail. Hence, we choose the index $q = 2 \in N$ *with* $\zeta_q = 5 > 0$ *as the pivot column (we could choose* $q = 1 \in N$ *as well) and compute*

$$\frac{\beta_3}{\gamma_{32}} = \frac{24}{4} = 6, \quad \frac{\beta_4}{\gamma_{42}} = \frac{21}{1} = 21, \quad \frac{\beta_5}{\gamma_{52}} = \frac{9}{1} = 9.$$

The first fraction achieves the minimal value and thus defines the pivot row to be $p = 3 \in B$.

Iteration 1:

The basis change leads to $B = \{2,4,5\}$ *and* $N = \{1,3\}$ *(we interchanged the pivot column* $q = 2$ *and the pivot row* $p = 3$*) and we find*

$$A_B = \begin{pmatrix} 4 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \; A_N = \begin{pmatrix} 1 & 1 \\ 3 & 0 \\ 1 & 0 \end{pmatrix}, \; c_B = \begin{pmatrix} -5 \\ 0 \\ 0 \end{pmatrix}, \; c_N = \begin{pmatrix} -2 \\ 0 \end{pmatrix}$$

and

$$\begin{pmatrix} \gamma_{21} & \gamma_{23} \\ \gamma_{41} & \gamma_{43} \\ \gamma_{51} & \gamma_{53} \end{pmatrix} = \frac{1}{4}\begin{pmatrix} 1 & 1 \\ 11 & -1 \\ 3 & -1 \end{pmatrix}, \; \begin{pmatrix} \beta_2 \\ \beta_4 \\ \beta_5 \end{pmatrix} = \begin{pmatrix} 6 \\ 15 \\ 3 \end{pmatrix}, \; \begin{pmatrix} \zeta_1 \\ \zeta_3 \end{pmatrix} = \frac{1}{4}\begin{pmatrix} 3 \\ -5 \end{pmatrix}$$

The checks for optimality and unboundedness fail again. Hence, we have to choose the index $q = 1 \in N$ *with* $\zeta_q = 3/4$ *as the pivot column and compute*

$$\frac{\beta_2}{\gamma_{21}} = \frac{6}{1/4} = 24, \quad \frac{\beta_4}{\gamma_{41}} = \frac{15}{11/4} = 60/11, \quad \frac{\beta_5}{\gamma_{51}} = \frac{3}{3/4} = 4.$$

The third fraction achieves the minimal value and thus defines the pivot row to be $p = 5 \in B$.

Iteration 2:

The basis change leads to $B = \{2, 4, 1\}$ and $N = \{5, 3\}$ (we interchanged the pivot column $q = 1$ and the pivot row $p = 5$) and we find

$$A_B = \begin{pmatrix} 4 & 0 & 1 \\ 1 & 1 & 3 \\ 1 & 0 & 1 \end{pmatrix}, \; A_N = \begin{pmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{pmatrix}, \; c_B = \begin{pmatrix} -5 \\ 0 \\ -2 \end{pmatrix}, \; c_N = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

and

$$\begin{pmatrix} \gamma_{25} & \gamma_{23} \\ \gamma_{45} & \gamma_{43} \\ \gamma_{15} & \gamma_{13} \end{pmatrix} = \frac{1}{3}\begin{pmatrix} -1 & 1 \\ -11 & 2 \\ 4 & -1 \end{pmatrix}, \; \begin{pmatrix} \beta_2 \\ \beta_4 \\ \beta_1 \end{pmatrix} = \begin{pmatrix} 5 \\ 4 \\ 4 \end{pmatrix}, \; \begin{pmatrix} \zeta_5 \\ \zeta_3 \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$$

Now, the optimality check does not fail as $\zeta_j < 0$ holds for all $j \in N$. Hence, the current feasible basic solution

$$x_B = \begin{pmatrix} x_2 \\ x_4 \\ x_1 \end{pmatrix} = \begin{pmatrix} 5 \\ 4 \\ 4 \end{pmatrix} = \beta, \qquad x_N = \begin{pmatrix} x_5 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

is optimal. The objective function value is $d = c_B^\top \beta = -33$.

2.3 The Simplex Table

The computations in step (2)(i) of Algorithm 2.2.7 are quite time consuming, especially if an LP has to be solved by hand. Therefore, the simplex method often is given in a more compact notation – the simplex table, which is just a more efficient way to perform the computations in Algorithm 2.2.7. The relations

$$\begin{aligned} x_B &= \beta - \Gamma x_N, \\ c^\top x &= d - \zeta^\top x_N, \end{aligned}$$

compare (2.1) and (2.2), are combined in the following table:

	x_N	
x_B	$\Gamma = (\gamma_{ij}) := A_B^{-1} A_N$	$\beta := A_B^{-1} b$
	$\zeta^\top := c_B^\top A_B^{-1} A_N - c_N^\top$	$d := c_B^\top A_B^{-1} b$

As the non-basic variable is zero, the current value of the variable x can be immediately obtained from the table: $x_B = \beta$, $x_N = 0$. The corresponding objective function is $c^\top x = d$.

Example 2.3.1 (compare Example 2.2.9)
Consider the standard LP

$$\text{Minimise} \quad c^\top x \quad \text{subject to} \quad Ax = b,\ x \geq 0$$

with the data (x_3, x_4, x_5 are slack variables):

$$c = \begin{pmatrix} -2 \\ -5 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \qquad A = \begin{pmatrix} 1 & 4 & 1 & 0 & 0 \\ 3 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}, \qquad b = \begin{pmatrix} 24 \\ 21 \\ 9 \end{pmatrix}, \qquad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix}.$$

A feasible basic solution is given by the basis index set $B = \{3, 4, 5\}$ and non-basis index set $N = \{1, 2\}$. Moreover, $A_B = I$ and

$$\begin{aligned}
\beta &= A_B^{-1} b = \begin{pmatrix} 24 \\ 21 \\ 9 \end{pmatrix}, \\
\Gamma &= A_B^{-1} A_N = \begin{pmatrix} 1 & 4 \\ 3 & 1 \\ 1 & 1 \end{pmatrix}, \\
\zeta^\top &= c_B^\top A_B^{-1} A_N - c_N^\top = \begin{pmatrix} 2 & 5 \end{pmatrix}, \\
d &= c_B^\top A_B^{-1} b = 0.
\end{aligned}$$

Hence, it holds

$$\begin{aligned}
x_B = \begin{pmatrix} x_3 \\ x_4 \\ x_5 \end{pmatrix} &= \begin{pmatrix} 24 \\ 21 \\ 9 \end{pmatrix} - \begin{pmatrix} 1 & 4 \\ 3 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\
c^\top x = d - \zeta^\top x_N &= 0 - \begin{pmatrix} 2 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.
\end{aligned}$$

The corresponding table is given by

	x_1	x_2	
x_3	1	4	24
x_4	3	1	21
x_5	1	1	9
	2	5	0

The simplex method requires to solve many linear equations in order to compute Γ and β after each basis change. This effort can be reduced by so-called update formulae for Γ, β, ζ, and d. Our intention is to derive an updated simplex table after a basis change has been performed. The new basis leads to a similar representation

$$\begin{aligned} x_{B^+} &= \beta^+ - \Gamma^+ x_{N^+}, \\ c^\top x^+ &= d^+ - {\zeta^+}^\top x_{N^+}. \end{aligned}$$

Our aim is to compute Γ^+, β^+, ζ^+, and d^+ without solving linear equations explicitly. For a basis index set B and the non-basis index set N, it holds

$$\begin{aligned} x_i &= \beta_i - \sum_{j\in N} \gamma_{ij} x_j, \qquad i \in B, \\ c^\top x &= d - \sum_{j\in N} \zeta_j x_j. \end{aligned}$$

Let $p \in B$ and $q \in N$ be the pivot row and the pivot column, respectively, and $\gamma_{pq} \neq 0$ the pivot element. Furthermore, let $B^+ = (B \setminus \{p\}) \cup \{q\}$ and $N^+ = (N \setminus \{q\}) \cup \{p\}$ be the new basis and non-basis index sets after a basis change.
Consider row p in the table:

$$x_p = \beta_p - \sum_{j\in N} \gamma_{pj} x_j = \beta_p - \gamma_{pq} x_q - \sum_{j\in N, j\neq q} \gamma_{pj} x_j.$$

Since $\gamma_{pq} \neq 0$, we can solve this equation for x_q and obtain

$$x_q = \frac{1}{\gamma_{pq}} \left(\beta_p - x_p - \sum_{j\in N, j\neq q} \gamma_{pj} x_j \right) = \underbrace{\frac{\beta_p}{\gamma_{pq}}}_{=\beta_q^+} - \underbrace{\frac{1}{\gamma_{pq}}}_{=\gamma_{qp}^+} x_p - \sum_{j\in N, j\neq q} \underbrace{\frac{\gamma_{pj}}{\gamma_{pq}}}_{=\gamma_{qj}^+} x_j.$$

Introducing x_q into the remaining equations

$$x_i = \beta_i - \sum_{j\in N} \gamma_{ij} x_j = \beta_i - \gamma_{iq} x_p - \sum_{j\in N, j\neq q} \gamma_{ij} x_j, \quad i \neq p,\ i \in B,$$

yields

$$x_i = \underbrace{\left(\beta_i - \gamma_{iq} \frac{\beta_p}{\gamma_{pq}} \right)}_{=\beta_i^+} - \underbrace{\left(-\frac{\gamma_{iq}}{\gamma_{pq}} \right)}_{=\gamma_{ip}^+} x_p - \sum_{j\in N, j\neq q} \underbrace{\left(\gamma_{ij} - \gamma_{iq} \frac{\gamma_{pj}}{\gamma_{pq}} \right)}_{=\gamma_{ij}^+} x_j, \quad i \neq p,\ i \in B.$$

Introducing x_q into the objective function yields

$$\begin{aligned} c^\top x &= d - \sum_{j\in N} \zeta_j x_j = d - \zeta_q x_q - \sum_{j\in N, j\neq q} \zeta_j x_j \\ &= \underbrace{d - \zeta_q \frac{\beta_p}{\gamma_{pq}}}_{=d^+} - \underbrace{\left(-\frac{\zeta_q}{\gamma_{pq}}\right)}_{=\zeta_p^+} x_p - \sum_{j\in N, j\neq q} \underbrace{\left(\zeta_j - \zeta_q \frac{\gamma_{pj}}{\gamma_{pq}}\right)}_{=\zeta_j^+} x_j \end{aligned}$$

Hence, the new basis is represented by

$$\begin{aligned} x_{B^+} &= \beta^+ - \Gamma^+ x_{N^+}, \\ c^\top x^+ &= d^+ - {\zeta^+}^\top x_{N^+}, \end{aligned}$$

where Γ^+, β^+, ζ^+, and d^+ are obtained by updating the values of Γ, β, ζ, and d according to the following update rules: For $i \in B^+$ and $j \in N^+$ it holds

$$\begin{aligned} \gamma_{ij}^+ &= \begin{cases} \gamma_{ij} - \frac{\gamma_{iq}\gamma_{pj}}{\gamma_{pq}} & \text{for } i \neq q, j \neq p \\ -\frac{\gamma_{iq}}{\gamma_{pq}} & \text{for } i \neq q, j = p \\ \frac{\gamma_{pj}}{\gamma_{pq}} & \text{for } i = q, j \neq p \\ \frac{1}{\gamma_{pq}} & \text{for } i = q, j = p \end{cases} \\ \beta_i^+ &= \begin{cases} \beta_i - \frac{\gamma_{iq}\beta_p}{\gamma_{pq}} & \text{for } i \neq q \\ \frac{\beta_p}{\gamma_{pq}} & \text{for } i = q \end{cases} \\ \zeta_j^+ &= \begin{cases} \zeta_j - \frac{\zeta_q\gamma_{pj}}{\gamma_{pq}} & \text{for } j \neq p \\ -\frac{\zeta_q}{\gamma_{pq}} & \text{for } j = p \end{cases} \\ d^+ &= d - \frac{\zeta_q\beta_p}{\gamma_{pq}} \end{aligned} \tag{2.11}$$

The updated simplex table after a basis change is given by:

	$x_j, j \in N \setminus \{q\}$	x_p	
$x_i, i \in B \setminus \{p\}$	$\gamma_{ij} - \frac{\gamma_{iq}\gamma_{pj}}{\gamma_{pq}}$	$-\frac{\gamma_{iq}}{\gamma_{pq}}$	$\beta_i - \frac{\gamma_{iq}\beta_p}{\gamma_{pq}}$
x_q	$\frac{\gamma_{pj}}{\gamma_{pq}}$	$\frac{1}{\gamma_{pq}}$	$\frac{\beta_p}{\gamma_{pq}}$
	$\zeta_j - \frac{\zeta_q\gamma_{pj}}{\gamma_{pq}}$	$-\frac{\zeta_q}{\gamma_{pq}}$	$d - \frac{\zeta_q\beta_p}{\gamma_{pq}}$

Example 2.3.2 (compare Examples 2.2.5 and 2.2.9)
Consider the standard LP

$$\textit{Minimise} \quad c^\top x \quad \textit{subject to} \quad Ax = b,\ x \geq 0$$

with the data (x_3, x_4, x_5 are slack variables):

$$c = \begin{pmatrix} -2 \\ -5 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad A = \begin{pmatrix} 1 & 4 & 1 & 0 & 0 \\ 3 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 24 \\ 21 \\ 9 \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix}.$$

The simplex method yields the following simplex tables (an initial feasible basic solution was given by the basis index set $B = \{3, 4, 5\}$):

Initial table:

	x_1	x_2	
x_3	1	$\boxed{4}$	24
x_4	3	1	21
x_5	1	1	9
	2	5	0

Pivot row and column: $p = 3$, $q = 2$, pivot element: $\gamma_{pq} = 4$.

Table 1:

	x_1	x_3	
x_2	$\frac{1}{4}$	$\frac{1}{4}$	6
x_4	$\frac{11}{4}$	$-\frac{1}{4}$	15
x_5	$\boxed{\frac{3}{4}}$	$-\frac{1}{4}$	3
	$\frac{3}{4}$	$-\frac{5}{4}$	-30

Pivot row and column: $p = 5$, $q = 1$, pivot element: $\gamma_{pq} = \frac{3}{4}$.

Table 2:

	x_5	x_3	
x_2	$-\frac{1}{3}$	$\frac{1}{3}$	5
x_4	$-\frac{11}{3}$	$\frac{2}{3}$	4
x_1	$\frac{4}{3}$	$-\frac{1}{3}$	4
	-1	-1	-33

Table 2 is optimal. The optimal solution is $x_2 = 5$, $x_4 = 4$, $x_1 = 4$, $x_5 = 0$, and $x_3 = 0$. The optimal objective function value is -33.

Important: Please note that the contents of the tables coincide with the computations performed in Example 2.2.9. The rules for choosing the pivot element and the tests for optimality and unboundedness are precisely the same rules as in Algorithm 2.2.7!

Remark 2.3.3

The vector ζ in an optimal table indicates whether the solution is unique. If $\zeta < 0$ holds *in an optimal simplex table, then the solution of the LP is unique. If there exists a component* $\zeta_j = 0$, $j \in N$, *in an optimal simplex table, then the optimal solution may not be unique. Further optimal solutions can be computed by performing additional basis changes by choosing those pivot columns with* $\zeta_j = 0$, $j \in N$. *By investigation of all possible basis changes it is possible to compute all optimal vertices. According to part (b) of the Fundamental Theorem of Linear Programming every optimal solution can be expressed as a convex combination of these optimal vertices.*

2.4 Phase 1 of the Simplex Method

A feasible basic solution is required to start the simplex method. In many cases such a solution can be obtained as follows.

Theorem 2.4.1 **(Canonical Problems)**
Consider an LP in canonical form 1.2.1

$$\text{Minimise} \quad c^\top x \quad \text{subject to} \quad Ax \leq b,\ x \geq 0.$$

If $b \geq 0$ *holds, then a feasible basic solution is given by*

$$y = b \geq 0, \quad x = 0,$$

where y *denotes the vector of slack variables. An initial simplex table is given by*

	x	
y	A	b
	$-c$	0

Proof: Introducing slack variables leads to a problem in standard form

$$Ax + Iy = b, \quad x \geq 0,\ y \geq 0.$$

As the unity matrix I is non-singular, a feasible basic solution is given by

$$y = b \geq 0, \quad x = 0.$$

□

Caution: If there exists a component $b_i < 0$, then it is much more complicated to find a feasible basic solution. The same holds true if the problem is not given in canonical form.

In all these cases, the following method has to be applied.

Consider an LP in standard form 1.2.5

$$\text{Minimise} \quad c^\top x \quad \text{subject to} \quad Ax = b,\ x \geq 0.$$

Without loss of generality we may assume $b \geq 0$. This is not a restriction as this property can always be achieved by multiplying by -1 those equations with $b_i < 0$.
Define

Auxiliary LP:

$$\text{Minimise} \quad e^\top y = \sum_{i=1}^{m} y_i \quad \text{subject to} \quad Ax + Iy = b,\ x \geq 0,\ y \geq 0, \tag{2.12}$$

where $e = (1, \ldots, 1)^\top \in \mathbb{R}^m$.

Theorem 2.4.2 (Feasible solution for auxiliary LP)
Consider the auxiliary LP 2.12 with $b \geq 0$. Then, a feasible basic solution is given by

$$y = b \geq 0, \quad x = 0.$$

An initial simplex table for the auxiliary LP is given by

	x	
y	A	b
	$e^\top A$	$e^\top b$

Proof: The auxiliary LP is a standard LP for $z = (x, y)^\top$. Obviously, it holds $c = (0, e^\top)^\top$ and $y = b - Ax$ and thus $\Gamma = A$ and $\beta = b$. Moreover, $c_B = e$ and $c_N = 0$. Hence, $\zeta^\top = c_B^\top \Gamma - c_N^\top = e^\top A$ and $d = c_B^\top \beta = e^\top b$. □

Theorem 2.4.3

(i) *If the auxiliary LP has the optimal solution $y = 0$, then the corresponding x obtained in the simplex method is a feasible basic solution for the standard LP 1.2.5.*

(ii) *If the auxiliary LP has the optimal solution $y \neq 0$ and thus $e^\top y > 0$, then the standard LP 1.2.5 is infeasible.*

Proof:

(i) If $y = 0$ is a solution, the corresponding x in the final simplex tableau satisfies $Ax = b$, $x \geq 0$, and thus is a feasible basic solution for the standard LP.

(ii) The objective function of the auxiliary LP is bounded from below by 0 because of $y \geq 0$ implies $e^\top y \geq 0$. Assume that the standard LP has a feasible point x. Then, $y = 0$ is feasible for the auxiliary LP because $Ax + y = b$, $x \geq 0$ and $y = 0$. Hence, $y = 0$ solves the auxiliary problem which contradicts the assumption in (ii).

□

Example 2.4.4 (Finding a feasible basic solution)
Consider the feasible set $Ax = b$, $x \geq 0$ with the data

$$A = \begin{pmatrix} 1 & 1 & 2 \\ -1 & 1 & 0 \end{pmatrix}, \qquad b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \qquad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}.$$

Notice that it is difficult to find a feasible basic solution at a first glance. Hence, we solve the following auxiliary problem by the simplex method.

$$\text{Minimise} \quad y_1 + y_2 \quad \text{subject to} \quad Ax + Iy = b, \quad x \geq 0, \quad y = (y_1, y_2)^\top \geq 0.$$

We obtain the following solution (notice that $x_4 := y_1$ and $x_5 := y_2$ in the computation):
Initial table:

	x_1	x_2	x_3	
x_4	1	1	2	1
x_5	−1	$\boxed{1}$	0	1
	0	2	2	2

Pivot row and column: $p = 5$, $q = 2$, pivot element: $\gamma_{pq} = 1$.
Table 1:

	x_1	x_5	x_3	
x_4	$\boxed{2}$	−1	2	0
x_2	−1	1	0	1
	2	−2	2	0

Pivot row and column: $p = 4$, $q = 1$, pivot element: $\gamma_{pq} = 2$.
Table 2:

	x_4	x_5	x_3	
x_1	$\frac{1}{2}$	$-\frac{1}{2}$	1	0
x_2	$\frac{1}{2}$	$\frac{1}{2}$	1	1
	−1	−1	0	0

This table is optimal with objective function value 0 *and* $y_1 = y_2 = 0 (= x_4 = x_5)$*. Hence,* $x = (x_1, x_2, x_3)^\top = (0, 1, 0)^\top$ *satisfies the constraints* $Ax = b$ *and* $x \geq 0$*.* x *is a feasible basic solution with basis index set* $B = \{1, 2\}$ *and non-basis index set* $N = \{3\}$*.*
Suppose now that we want to minimise the function

$$x_1 + 2x_2 + x_3 \qquad (\text{i.e. } c = (1, 2, 1)^\top)$$

on the feasible set above. The initial simplex table is given by table 2 if the columns corresponding to the auxiliary variables y_1 *and* y_2 *(resp.* x_4 *and* x_5*) are deleted. As the objective function has changed, we have to recompute the bottom row of the table, that is*

$$\begin{aligned} d = c_B^\top \beta &= (1, 2) \begin{pmatrix} 0 \\ 1 \end{pmatrix} = 2, \\ \zeta^\top &= c_B^\top \Gamma - c_N^\top = (1, 2) \begin{pmatrix} 1 \\ 1 \end{pmatrix} - (1) = 2. \end{aligned}$$

Initial table:

	x_3	
x_1	$\boxed{1}$	0
x_2	1	1
	2	2

Pivot row and column: $p = 1$*,* $q = 3$*, pivot element:* $\gamma_{pq} = 1$*.*
Table 1:

	x_1	
x_3	1	0
x_2	-1	1
	-2	2

This table is optimal.

Remark 2.4.5

In might happen that some components of y *are among the basic variables in the final simplex table. In this case the columns corresponding to the components* x_i*,* $i \in B$*, do not define a full basis matrix (always* m *linearly independent columns are necessary). Hence, additional linearly independent columns* a^j *of* A *with* $j \in N$ *have to be determined in order to obtain a complete basis matrix. This can be done by inspection or by performing additional simplex steps with the aim to move the auxiliary variable* y *into the non-basis. This can be achieved by performing steps of the dual simplex method, compare Chapter 3.*

2.5 Computational Issues

Our examples so far suggest that the simplex method seems to work properly. However, there are some pitfalls which were encountered during the decades after the invention of the method.

Open questions in 1947:

- If the pivot element is not uniquely determined, which one should be chosen? There may exist several choices for p and q!
- Finiteness: Does the algorithm always terminate?
- Complexity: polynomial effort or non-polynomial (i.e. exponential) effort?

2.5.1 Finite Termination of the Simplex Method

There are two degrees of freedom in step (4) of Algorithm 2.2.7:

- Which index q with $\zeta_q > 0$ should be chosen? The examples 2.2.9, 2.3.2, 2.4.4 show, that there are often many choices.
- Which index p with
$$\frac{\beta_p}{\gamma_{pq}} = \min\left\{ \frac{\beta_i}{\gamma_{iq}} \;\middle|\; \gamma_{iq} > 0,\ i \in B \right\}$$
should be chosen? In general there may exist more than one choice, e.g. in the initial table of Example 2.4.4.

Dantzig has chosen the pivot column according to the formula

$$\zeta_q = \max\{\zeta_j \mid \zeta_j > 0,\ j \in N\}, \tag{2.13}$$

because this choice guarantees the largest descent in the objective function if the constraints are neglected. Unfortunately, this choice may lead to cycles in the simplex method as famous examples by Hoffman [Hof53] and Marshall and Suurballe [MS69] show. In this case, the simplex method does not terminate!

Example 2.5.1 **(Cycles in simplex method)**
Consider the example of Marshall and Suurballe [MS69]:

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

with the data (x_5, x_6, x_7 are slack variables)

$$c = \begin{pmatrix} -10 \\ 57 \\ 9 \\ 24 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad A = \begin{pmatrix} 0.5 & -5.5 & -2.5 & 9 & 1 & 0 & 0 \\ 0.5 & -1.5 & -0.5 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix}.$$

With Dantzig's choice of the pivot column, the simplex method yields the following tables. Notice that the last table is the same as the first table (apart from perturbations of the columns). Hence, a similar choice of the pivot columns leads to a cycle and the simplex method does not terminate.

Initial table:

	x_1	x_2	x_3	x_4	
x_5	**[0.5]**	-5.5	-2.5	9	0
x_6	0.5	-1.5	-0.5	1	0
x_7	1	0	0	0	1
	10	-57	-9	-24	0

Pivot row and column: $p = 5$, $q = 1$, pivot element: $\gamma_{pq} = 0.5$.

Table 1:

	x_5	x_2	x_3	x_4	
x_1	2	-11	-5	18	0
x_6	-1	**[4]**	2	-8	0
x_7	-2	11	5	-18	1
	-20	53	41	-204	0

Pivot row and column: $p = 6$, $q = 2$, pivot element: $\gamma_{pq} = 4$.

Table 2:

	x_5	x_6	x_3	x_4	
x_1	-0.75	2.75	**[0.5]**	-4	0
x_2	-0.25	0.25	0.5	-2	0
x_7	0.75	-2.75	-0.5	4	1
	-6.75	-13.25	14.5	-98	0

Pivot row and column: $p = 1$, $q = 3$, pivot element: $\gamma_{pq} = 0.5$.

Table 3:

	x_5	x_6	x_1	x_4	
x_3	-1.5	5.5	2	-8	0
x_2	0.5	-2.5	-1	$\boxed{2}$	0
x_7	0	0	1	0	1
	15	-93	-29	18	0

Pivot row and column: $p = 2$, $q = 4$, *pivot element:* $\gamma_{pq} = 2$.

Table 4:

	x_5	x_6	x_1	x_2	
x_3	$\boxed{0.5}$	-4.5	-2	4	0
x_4	0.25	-1.25	-0.5	0.5	0
x_7	0	0	1	0	1
	10.5	-70.5	-20	-9	0

Pivot row and column: $p = 3$, $q = 5$, *pivot element:* $\gamma_{pq} = 0.5$.

Table 5:

	x_3	x_6	x_1	x_2	
x_5	2	-9	-4	8	0
x_4	-0.5	$\boxed{1}$	0.5	-1.5	0
x_7	0	0	1	0	1
	-21	24	22	-93	0

Pivot row and column: $p = 4$, $q = 6$, *pivot element:* $\gamma_{pq} = 1$.

Table 6:

	x_3	x_4	x_1	x_2	
x_5	-2.5	9	0.5	-5.5	0
x_6	-0.5	1	0.5	-1.5	0
x_7	0	0	1	0	1
	-9	-24	10	-57	0

Example 2.5.1 reveals an interesting observation: Although all tables are different (with exception of the initial and final table), the corresponding basic solutions are always the same, namely

$$x = (0, 0, 0, 0, 0, 0, 1)^\top.$$

With other words: All tables describe the same vertex x. This clearly shows that the same vertex may possess different table representations and in this example we unfortunately produced a cycle.

Moreover, we observe that two basic components of the feasible basic solution are always zero in all tables, e.g. in table 3 the basic variables x_2 and x_3 are zero. Whenever this happens, we call the basic solution degenerated:

Definition 2.5.2 **(Degenerate feasible basic solution)**
Let $x = (x_1, \ldots, x_n)^\top \in \mathbb{R}^n$ be a feasible basic solution with basis index set B. x is called degenerate, if $x_i = 0$ holds for at least one index $i \in B$. If $x_i > 0$ for all $i \in B$, x is called non-degenerate.

We now show why degeneracy and cycling are closely connected.

The non-degenerate case: Assume that the simplex method always chooses pivot rows p with $\beta_p > 0$. Then the transformation rules (2.11) show that the objective function value strictly decreases in this case because

$$d^+ = d - \frac{\overbrace{\zeta_q}^{>0} \overbrace{\beta_p}^{>0}}{\underbrace{\gamma_{pq}}_{>0}} < d.$$

As the simplex method by construction computes feasible basic solutions only and as there exist only finitely many of them, one of them being optimal, the simplex method terminates in a finite number of steps (provided that a solution exists at all).

The degenerate case: The non-degenerate case shows that a cycle can only occur if a pivot row p with $\beta_p = 0$ has to be chosen in the simplex method. In this case it holds

$$d^+ = d - \frac{\overbrace{\zeta_q}^{>0} \overbrace{\beta_p}^{=0}}{\underbrace{\gamma_{pq}}_{>0}} = d.$$

Hence, the objective function value remains unchanged in a basis change from x to x^+. Moreover, the transformation rules (2.11) yield

$$\beta_i^+ = \begin{cases} \beta_i - \frac{\gamma_{iq}}{\gamma_{pq}} \overbrace{\beta_p}^{=0} & \text{for } i \neq q \\ \frac{\overbrace{\beta_p}^{=0}}{\gamma_{pq}} & \text{for } i = q \end{cases} = \begin{cases} \beta_i & \text{for } i \neq q \\ 0 & \text{for } i = q \end{cases}$$

Hence, $\beta^+ = \beta$ and thus $x_{B^+} = \beta^+ = \beta = x_B$ (compare Example 2.5.1). The simplex method stagnates in the feasible basic solution $x = x^+$. Notice that the simplex table does change although x remains the same. Now, Example 2.5.1 shows that it is actually possible to revisit a simplex table again after a finite number of basis changes. If this happens, a cycle is born.

Summarising:

- A cycle can only occur in a degenerate feasible basic solution.
- Example 2.4.4 shows that the presence of a degenerate feasible basic solution does not automatically lead to a cycle.

Can cycles be avoided at all? Fortunately yes. There is a simple rule (more precisely: the rule is easy to apply but very difficult to prove) that avoids cycling.

Bland's rule [Bla77]

Among all possible choices for the pivot element, always choose the pivot column $q \in N$ and the pivot row $p \in B$ in step (4) of Algorithm 2.2.7 with the smallest indices q and p, that is:

(i) Choose $q = \min\{j \in N \mid \zeta_j > 0\}$.

(ii) Choose $p = \min\left\{ k \in B \;\middle|\; \frac{\beta_k}{\gamma_{kq}} = \min\left\{ \frac{\beta_i}{\gamma_{iq}} \;\middle|\; \gamma_{iq} > 0,\ i \in B \right\} \right\}$.

Finally, we state the main result of this section:

Theorem 2.5.3 **(Finite termination)**

If the pivot element is chosen according to Bland's rule, the Simplex Algorithm 2.2.7 either finds an optimal solution or it detects that the objective function is unbounded from below. In both cases Algorithm 2.2.7 terminates after a finite number of steps.

Remark 2.5.4

In practical applications cycling usually does not occur (due to round-off errors) and often Bland's rule is not obeyed. Instead, Dantzig's rule of steepest descent, compare equation (2.13), is used frequently (although – from a theoretical point of view – it may lead to cycles).

2.5.2 Complexity of the Simplex Method

The previous paragraph showed that the simplex method terminates after a finite number of steps if Bland's rule is used. This is a good message. However, in view of the numerical performance of an algorithm the most relevant question is that of its complexity.

Example 2.5.5 **(Polynomial and exponential complexity)**

Suppose a computer with 10^{10} ops/sec (10 GHz) is given. A problem P (e.g. an LP) has to be solved on the computer. Let n denote the size of the problem (e.g. the number of variables of the LP). Let five different algorithms be given, which require n, n^2, n^4, 2^n,

and $n!$ operations (e.g. steps in the simplex method), respectively, to solve the problem P of size n.
The following table provides an overview on the time which is needed to solve the problem.

ops \ size n	20	60	⋯	100	1000
n	0.002 μs	0.006 μs	⋯	0.01 μs	0.1 μs
n^2	0.04 μs	0.36 μs	⋯	1 μs	0.1 ms
n^4	16 μs	1.296 ms	⋯	10 ms	100 s
2^n	0.1 ms	3 yrs	⋯	10^{12} yrs	·
$n!$	7.7 yrs	·	·	·	·

Clearly, only the algorithms with polynomial complexity n, n^2, n^4 are 'efficient' while those with exponential complexity 2^n and $n!$ are very 'inefficient'.

We don't want to define the complexity of an algorithm in a mathematically sound way. Instead we use the following informal but intuitive definition:

Definition 2.5.6 (Complexity (informal definition))
A method for solving a problem P has polynomial complexity if it requires at most a polynomial (as a function of the size) number of operations for solving any instance of the problem P. A method has exponential complexity, if it has not polynomial complexity.

Unfortunately, it was shown by the example below of Klee and Minty [KM72] in 1972 that

the simplex method in its worst case has exponential complexity!

Hence, the simplex method does not belong to the class of algorithms with polynomial complexity. This was a very disappointing result.

Example 2.5.7 (Klee and Minty [KM72])
Klee and Minty showed that the following linear program has 2^n vertices and that there exists a simplex path which visits every 2^n vertices and thus, the simplex method requires an exponential number of steps w.r.t. n. Notice, that each step of the simplex method only requires a polynomial number of operations as essentially linear equations have to be solved.

$$\max \quad e_n^\top x \quad \textit{subject to} \quad 0 \le x_1 \le 1, \quad \varepsilon x_i \le x_{i+1} \le 1 - \varepsilon x_i, \quad i = 1, \ldots, n-1$$

with $\varepsilon \in (0, 1/2)$ and $e_n^\top = (1, \ldots, 1)$.

From a theoretical point of view, the simplex method is an inefficient method. Nevertheless, there is also a practical point of view. And the simplex method is still one of the most frequently used algorithms in linear programming, even for large-scale problems. Moreover, extensive numerical experience with practically relevant problems showed the

following: Most often, the number of variables n is between m (number of constraints) and $10m$ and the simplex method needs between m and $4m$ steps to find a solution. As each step of the simplex method needs a polynomial number of operations in n and m, the simplex method solves most practically relevant problems in polynomial time.
Hence: In practise the simplex method shows polynomial complexity in n and m!
Further investigations by Borgwardt [Bor87] show: If the input data A, b and c is reasonably distributed, then the expected effort of the simplex method is polynomial.

2.5.3 The Revised Simplex Method

The main effort of the simplex method 2.2.7 is to compute the following quantities:

$$\Gamma = A_B^{-1} A_N, \qquad \beta = A_B^{-1} b.$$

We achieved this in two ways:

- Explicit computation: An efficient implementation of the simplex method does not compute the matrix A_B^{-1} explicitely. Rather, in order to obtain the columns γ^j, $j \in N$, of Γ and the vector β, the linear equations

  $$A_B \gamma^j = a^j, \quad j \in N, \qquad A_B \beta = b,$$

  are solved by some suitable numerical method, e.g. LU-decomposition (Gaussian-Algorithm) or QR-decomposition or iterative solvers. In total, $n - m + 1$ linear equations of dimension m have to be solved for this approach.

- Table update: The update formulae (2.11) can be used to update the simplex table. The main effort is to update the $m \times (n - m)$-matrix Γ.

If n is much larger than m, that is $m << n$, then it is not efficient to update the whole simplex table, because a close investigation of the simplex algorithm reveals that essentially only the vectors β, ζ and the pivot column γ^q of Γ are necessary to determine the pivot element γ_{pq}. The remaining columns $j \in N$, $j \neq q$, of Γ are not referenced at all in the algorithm. In fact it is more efficient (in view of computational effort and memory requirements) to work with the $m \times m$-matrix A_B only.

Algorithm 2.5.8 (Revised Simplex Method)

(0) Phase 0:

Transform the linear program into standard form 1.2.5, if necessary at all.

(1) Phase 1:

Determine a feasible basic solution (feasible vertex) x for the standard LP 1.2.5 with basis index set B, non-basis index set N, basis matrix A_B, basic variable $x_B \geq 0$ and non-basic variable $x_N = 0$.

If no feasible solution exists, STOP. The problem is infeasible.

(2) Phase 2:

(i) Compute $\beta = (\beta_i)_{i\in B}$ as the solution of the linear equation $A_B\beta = b$.

(ii) Solve the linear equation $A_B^\top \lambda = c_B$ for $\lambda \in \mathbb{R}^m$ and compute $\zeta = (\zeta_j)_{j\in N}$ by

$$\zeta^\top = \lambda^\top A_N - c_N^\top.$$

(ii) Check for optimality:
If $\zeta_j \leq 0$ for every $j \in N$, then STOP. The current feasible basic solution $x_B = \beta$, $x_N = 0$ is optimal. The objective function value is $d = c_B^\top \beta$.

(iii) Determine pivot column: Choose an index q with $\zeta_q > 0$ (according to Bland's rule).

(iv) Compute pivot column: Solve the linear equation $A_B\gamma = a^q$, where a^q denotes column q of A.

(v) Check for unboundedness:
If $\gamma \leq 0$, then the linear program does not have a solution and the objective function is unbounded from below. STOP.

(vi) Determine pivot row:
Choose an index p (according to Bland's rule) with

$$\frac{\beta_p}{\gamma_p} = \min\left\{ \frac{\beta_i}{\gamma_i} \;\middle|\; \gamma_i > 0,\ i \in B \right\}.$$

(vii) Perform basis change:
Set $B := (B \setminus \{p\}) \cup \{q\}$ and $N := (N \setminus \{q\}) \cup \{p\}$.

(viii) Go to (i).

The revised simplex algorithm requires to solve three linear equations only, namely two linear equations with A_B in steps (i) and (iv) and one linear equation with $A_B^\top$ in (ii):

$$A_B\beta = b, \qquad A_B\gamma = a^q, \qquad A_B^\top\lambda = c_B.$$

Solving these linear equations can be done using the following techniques:

- explicit computation of A_B^{-1} and computation of

$$\beta = A_B^{-1}b, \qquad \gamma = A_B^{-1}a^q, \qquad \lambda = (A_B^{-1})^\top c_B.$$

- Decomposition of A_B using an LU-decomposition. Herein, the matrix A_B is expressed as $A_B = L \cdot U$ with a lower triangular matrix $L \in \mathbb{R}^{m\times m}$ and an upper triangular matrix $U \in \mathbb{R}^{m\times m}$. The linear equation $A_B\beta = b$ is then solved by forward-backward substitution. For details please refer to courses on numerical linear algebra or numerical mathematics.

- It can be exploited, that the basis matrices A_B and A_{B+} are neighbouring matrices. This allows to construct update formulae similar to (2.11) for the inverse basis matrix A_B^{-1}. Let $B = \{i_1, \ldots, i_{\ell-1}, p, i_{\ell+1}, \ldots, i_m\}$ with $1 \leq \ell \leq m$ basis index set and $\gamma = A_B^{-1} a^q$ with $q \in N$ the current pivot column with pivot element $\gamma_\ell \neq 0$. Herein, a^q is the q-th column of A. Let e_ℓ be the l-th unity vector. Then the following update rule holds:

$$A_{B+}^{-1} = \left(I - (\gamma - e_\ell)e_\ell^\top / \gamma_\ell\right) A_B^{-1} \tag{2.14}$$

Example 2.5.9 **(Revised Simplex Method)**

Minimise $-3x_1 - 4x_2$ *subject to*

$$\begin{aligned} 2x_1 + x_2 + x_3 &= 8, \\ 4x_1 + x_2 + x_4 &= 10, \\ x_1, x_2, x_3, x_4 &\geq 0. \end{aligned}$$

The problem is in standard form already with

$$c = \begin{pmatrix} -3 \\ -4 \\ 0 \\ 0 \end{pmatrix}, \qquad b = \begin{pmatrix} 8 \\ 10 \end{pmatrix}, \qquad A = \begin{pmatrix} 2 & 1 & 1 & 0 \\ 4 & 1 & 0 & 1 \end{pmatrix}.$$

Phase 1: A feasible basic solution is given by $x_B = (8, 10)^\top$, $x_N = 0$ *with* $B = \{3, 4\}$ *and* $N = \{1, 2\}$.

Phase 2:

Iteration 0:

With $B = \{3, 4\}$ *and* $N = \{1, 2\}$ *we find*

$$A_B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \; A_N = \begin{pmatrix} 2 & 1 \\ 4 & 1 \end{pmatrix}, \; c_B = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \; c_N = \begin{pmatrix} -3 \\ -4 \end{pmatrix}.$$

and

$$\begin{pmatrix} \beta_3 \\ \beta_4 \end{pmatrix} = \begin{pmatrix} 8 \\ 10 \end{pmatrix}, \; \lambda = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \; \begin{pmatrix} \zeta_1 \\ \zeta_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \end{pmatrix}.$$

The check for optimality fails. Hence, in accordance with Bland's rule we choose the index $q = 1 \in N$ *with* $\zeta_q = 3 > 0$ *as the pivot column. The pivot column computes to*

$$\begin{pmatrix} \gamma_3 \\ \gamma_4 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \end{pmatrix}.$$

The check for unboundedness fails. The pivot row is $p = 4 \in B$, because

$$\frac{\beta_3}{\gamma_3} = \frac{8}{2} = 4, \quad \frac{\beta_4}{\gamma_4} = \frac{10}{4} = 2.5.$$

Iteration 1:

The basis change leads to $B = \{3, 1\}$ and $N = \{4, 2\}$ (we interchanged the pivot column $q = 1$ and the pivot row $p = 4$) and we find

$$A_B = \begin{pmatrix} 1 & 2 \\ 0 & 4 \end{pmatrix}, \; A_N = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, \; c_B = \begin{pmatrix} 0 \\ -3 \end{pmatrix}, \; c_N = \begin{pmatrix} 0 \\ -4 \end{pmatrix}$$

and

$$\begin{pmatrix} \beta_3 \\ \beta_1 \end{pmatrix} = \begin{pmatrix} 3 \\ 2.5 \end{pmatrix}, \; \lambda = \begin{pmatrix} 0 \\ -\frac{3}{4} \end{pmatrix}, \; \begin{pmatrix} \zeta_4 \\ \zeta_2 \end{pmatrix} = \frac{1}{4} \begin{pmatrix} -3 \\ 13 \end{pmatrix}.$$

The check for optimality fails. Hence, in accordance with Bland's rule we choose the index $q = 2 \in N$ with $\zeta_q = \frac{13}{4} > 0$ as the pivot column. The pivot column computes to

$$\begin{pmatrix} \gamma_3 \\ \gamma_1 \end{pmatrix} = \frac{1}{4} \begin{pmatrix} 2 \\ 1 \end{pmatrix}.$$

The check for unboundedness fails. The pivot row is $p = 3 \in B$.

Iteration 2:

The basis change leads to $B = \{2, 1\}$ and $N = \{4, 3\}$ (we interchanged the pivot column $q = 2$ and the pivot row $p = 3$) and we find

$$A_B = \begin{pmatrix} 1 & 2 \\ 1 & 4 \end{pmatrix}, \; A_N = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \; c_B = \begin{pmatrix} -4 \\ -3 \end{pmatrix}, \; c_N = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

and

$$\begin{pmatrix} \beta_2 \\ \beta_1 \end{pmatrix} = \begin{pmatrix} 6 \\ 1 \end{pmatrix}, \; \lambda = \frac{1}{2} \begin{pmatrix} -13 \\ 5 \end{pmatrix}, \; \begin{pmatrix} \zeta_4 \\ \zeta_3 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 5 \\ -13 \end{pmatrix}.$$

The check for optimality fails. Hence, in accordance with Bland's rule we choose the index $q = 4 \in N$ with $\zeta_q = \frac{5}{2} > 0$ as the pivot column. The pivot column computes to

$$\begin{pmatrix} \gamma_2 \\ \gamma_1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} -2 \\ 1 \end{pmatrix}.$$

The check for unboundedness fails. The pivot row is $p = 1 \in B$.

Iteration 3:

The basis change leads to $B = \{2,4\}$ and $N = \{1,3\}$ (we interchanged the pivot column $q = 4$ and the pivot row $p = 1$) and we find

$$A_B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \quad A_N = \begin{pmatrix} 2 & 1 \\ 4 & 0 \end{pmatrix}, \quad c_B = \begin{pmatrix} -4 \\ 0 \end{pmatrix}, \quad c_N = \begin{pmatrix} -3 \\ 0 \end{pmatrix}$$

and

$$\begin{pmatrix} \beta_2 \\ \beta_4 \end{pmatrix} = \begin{pmatrix} 8 \\ 2 \end{pmatrix}, \quad \lambda = \begin{pmatrix} -4 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} \zeta_1 \\ \zeta_3 \end{pmatrix} = \begin{pmatrix} -5 \\ -4 \end{pmatrix}.$$

The optimality criterion is satisfied. The optimal solution is $x = (0, 8, 0, 2)^\top$ with objective function value $c^\top x = -32$.

Remark 2.5.10

The auxiliary variable λ in the revised simplex method is a so-called dual variable for the dual linear program. Dual problems are discussed in the following chapter.

Chapter 3

Duality and Sensitivity

This chapter provides an introduction to duality theory for linear programs and points out the connection between dual programs and sensitivity analysis for linear programs. Sensitivity analysis addresses the question of how optimal solutions depend on perturbations in the problem data (A, b, and c). Finally, the results from sensitivity analysis will be used to derive so-called shadow prices which play an important role in economy.
Dual linear programs are important because:

- Dual problems allow to determine lower bounds for the optimal objective function value of a primal minimisation problem.
- Dual problems allow to compute sensitivities of linear programs.
- Occasionally, the dual problem is easier to solve than the primal problem ($\rightarrow$ dual simplex method).

3.1 The Dual Linear Program

Before we state the dual problem of a standard LP in a formal way, we consider two motivations for the so-called dual problem.

Example 3.1.1

A company produces three products P_1, P_2, and P_3. The profit of the products per unit amounts to 10, 5, and 5.5 pounds, respectively. The production requires four raw materials B_1, B_2, B_3, and B_4. There are 1500, 200, 1200, and 900 units of these raw materials in stock. The amount of raw materials needed for the production of one unit of the product is given by the following table.

	P_1	P_2	P_3
B_1	30	10	50
B_2	5	0	3
B_3	20	10	50
B_4	10	20	30

Let x_i, $i = 1, 2, 3$, denote the amount of product P_i, $i = 1, 2, 3$, to be produced. The company aims at maximising the profit and hence solves the following so-called primal problem:

Maximise

$$10x_1 + 5x_2 + 5.5x_3$$

subject to

$$\begin{aligned} 30x_1 + 10x_2 + 50x_3 &\leq 1500, \\ 5x_1 + 3x_3 &\leq 200, \\ 20x_1 + 10x_2 + 50x_3 &\leq 1200, \\ 10x_1 + 20x_2 + 30x_3 &\leq 900, \\ x_1, x_2, x_3 &\geq 0. \end{aligned}$$

Assume now, that a second company offers to buy all raw materials from the first company. The second company offers a price of $\lambda_i \geq 0$ pounds for one unit of the raw material B_i, $i = 1, \ldots, 4$. Of course, the second company intends to minimise its total costs

$$1500\lambda_1 + 200\lambda_2 + 1200\lambda_3 + 900\lambda_4.$$

Moreover, the first company will only accept the offer of the second company if the resulting price for one unit of the product P_j is greater than or equal to the (not realised) profit c_j, $j = 1, 2, 3$, i.e. if

$$\begin{aligned} 30\lambda_1 + 5\lambda_2 + 20\lambda_3 + 10\lambda_4 &\geq 10, \\ 10\lambda_1 + 10\lambda_3 + 20\lambda_4 &\geq 5, \\ 50\lambda_1 + 3\lambda_2 + 50\lambda_3 + 30\lambda_4 &\geq 5.5, \end{aligned}$$

hold. Summarising, the second company has to solve the following linear program:
Minimise

$$1500\lambda_1 + 200\lambda_2 + 1200\lambda_3 + 900\lambda_4.$$

subject to

$$\begin{aligned} 30\lambda_1 + 5\lambda_2 + 20\lambda_3 + 10\lambda_4 &\geq 10, \\ 10\lambda_1 + 10\lambda_3 + 20\lambda_4 &\geq 5, \\ 50\lambda_1 + 3\lambda_2 + 50\lambda_3 + 30\lambda_4 &\geq 5.5, \\ \lambda_1, \lambda_2, \lambda_3, \lambda_4 &\geq 0. \end{aligned}$$

This problem is called the dual problem of the primal problem.
More generally, we may associate to the primal problem

$$\textit{Maximise} \quad c^\top x \quad \textit{subject to} \quad Ax \leq b, \ x \geq 0$$

the dual problem

$$\text{Minimise} \quad b^\top \lambda \quad \text{subject to} \quad A^\top \lambda \geq c, \ \lambda \geq 0.$$

Now we are interested in formulating a dual problem for the standard problem, which will be called primal problem in the sequel.

Definition 3.1.2 (Primal problem (P))
The standard LP

$$(P) \qquad \text{Minimise} \quad c^\top x \quad \text{subject to} \quad Ax = b, \ x \geq 0$$

with $A \in \mathbb{R}^{m\times n}$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ is called primal problem.

The dual problem to the primal problem (P) is defined as follows.

Definition 3.1.3 (Dual problem (D))
The linear program

$$(D) \qquad \text{Maximise} \quad b^\top \lambda \quad \text{subject to} \quad A^\top \lambda \leq c$$

is called the dual problem of (P).

Example 3.1.4
Let the primal LP be given by the following linear program:
Minimise $-3x_1 - 4x_2$ subject to

$$\begin{aligned} 2x_1 + x_2 + x_3 &= 8, \\ 4x_1 + x_2 + x_4 &= 10, \\ x_1, x_2, x_3, x_4 &\geq 0. \end{aligned}$$

The dual problem reads as:
Maximise $8\lambda_1 + 10\lambda_2$ subject to

$$\begin{aligned} 2\lambda_1 + 4\lambda_2 &\leq -3, \\ \lambda_1 + \lambda_2 &\leq -4, \\ \lambda_1 &\leq 0, \\ \lambda_2 &\leq 0. \end{aligned}$$

By application of the well-known transformation techniques, which allow to transform a general LP into a standard LP, and writing down the dual problem of the resulting standard problem, it is possible to formulate the dual problem of a general LP. We demonstrate this procedure for the following problem (compare Example 3.1.1). Please note the nice symmetry of the primal and dual problem!

Theorem 3.1.5

The dual problem of the LP

$$\text{Maximise} \quad c^\top x \quad \text{subject to} \quad Ax \leq b,\ x \geq 0$$

is given by

$$\text{Minimise} \quad b^\top \lambda \quad \text{subject to} \quad A^\top \lambda \geq c,\ \lambda \geq 0.$$

Proof: Transformation into standard form yields:

$$\text{Minimise} \quad \begin{pmatrix} -c^\top & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad \text{subject to} \quad \begin{pmatrix} A & I \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = b,\ \begin{pmatrix} x \\ y \end{pmatrix} \geq 0.$$

The dual problem of this standard problem reads as

$$\text{Maximise} \quad b^\top \tilde{\lambda} \quad \text{subject to} \quad \begin{pmatrix} A^\top \\ I \end{pmatrix} \tilde{\lambda} \leq \begin{pmatrix} -c \\ 0 \end{pmatrix}.$$

Defining $\lambda := -\tilde{\lambda}$ yields the problem

$$\text{Maximise} \quad -b^\top \lambda \quad \text{subject to} \quad -A^\top \lambda \leq -c,\ -\lambda \leq 0,$$

which proves the assertion. □

In a similar way, the following scheme applies. Notice in the scheme below, that the primal problem is supposed to be a minimisation problem. As we know already, this can always be achieved by multiplying the objective function of a maximisation problem by -1.

primal constraints (minimise $c^\top x$)	dual constraints (maximise $b^\top \lambda$)
$x \geq 0$	$A^\top \lambda \leq c$
$x \leq 0$	$A^\top \lambda \geq c$
x free	$A^\top \lambda = c$
$Ax = b$	λ free
$Ax \leq b$	$\lambda \leq 0$
$Ax \geq b$	$\lambda \geq 0$

The same scheme applies component-wise if primal constraints of the different types

$$x_i \left\{ \begin{matrix} \geq 0 \\ \leq 0 \\ \text{free} \end{matrix} \right\},\ i = 1, \ldots, n, \qquad \sum_{j=1}^{n} a_{ij} x_j \left\{ \begin{matrix} = \\ \leq \\ \geq \end{matrix} \right\} b_i,\ i = 1, \ldots, m,$$

occur simultaneously in a general LP. More precisely, the dual variable λ_i is associated with the i-th primal constraint according to the scheme

$$\sum_{j=1}^{n} a_{ij}x_j \left\{ \begin{array}{c} = \\ \leq \\ \geq \end{array} \right\} b_i \qquad \leftrightarrow \qquad \lambda_i \left\{ \begin{array}{c} \text{free} \\ \leq 0 \\ \geq 0 \end{array} \right\}.$$

Moreover, the i-th primal variable x_i is associated with the i-th component of the dual constraint according to the scheme

$$x_i \left\{ \begin{array}{c} \geq 0 \\ \leq 0 \\ \text{free} \end{array} \right\} \qquad \leftrightarrow \qquad (A^\top \lambda)_i \left\{ \begin{array}{c} \leq \\ \geq \\ = \end{array} \right\} c_i.$$

In the latter, $(A^\top \lambda)_i$ denotes the i-th component of the vector $A^\top \lambda$, where $A = (a_{ij})$ denotes the matrix of coefficients of the primal constraints (excluding sign conditions of course). Again, the above relations assume that the primal problem is a minimisation problem.

Remark 3.1.6

Dualisation of the dual problem yields the primal problem again.

3.2 Weak and Strong Duality

We summarise important relations between the primal problem and its dual problem.

Theorem 3.2.1 (Weak Duality Theorem)

Let x be feasible for the primal problem (P) (i.e. $Ax = b$, $x \geq 0$) and let λ be feasible for the dual problem (D) (i.e. $A^\top \lambda \leq c$). Then it holds

$$b^\top \lambda \leq c^\top x.$$

Proof: Owing to $b = Ax$ and $x \geq 0$ it holds

$$b^\top \lambda = (Ax)^\top \lambda = x^\top A^\top \lambda \leq x^\top c = c^\top x.$$

□

The weak duality theorem provides a motivation for the dual problem: dual feasible points provide lower bounds for the optimal objective function value of the primal problem. Vice versa, primal feasible points provide upper bounds for the optimal objective function value of the dual problem. This property is very important in the context of Branch & Bound methods for integer programs.

Moreover, it holds

Theorem 3.2.2 **(Sufficient Optimality Criterion)**

Let x be feasible for the primal problem (P) and let λ be feasible for the dual problem (D).

(i) If $b^\top\lambda = c^\top x$, then x is optimal for the primal problem (P) and λ is optimal for the dual problem (D).

(ii) $b^\top\lambda = c^\top x$ holds if and only if the complementarity slackness condition holds:

$$x_j \left(\sum_{i=1}^{m} a_{ij}\lambda_i - c_j \right) = 0, \qquad j = 1, \ldots, n.$$

Remark 3.2.3

The complementary slackness condition is equivalent with the following: For $j = 1, \ldots, n$ it holds

$$x_j > 0 \qquad \Rightarrow \qquad \sum_{i=1}^{m} a_{ij}\lambda_i - c_j = 0$$

and

$$\sum_{i=1}^{m} a_{ij}\lambda_i - c_j < 0 \qquad \Rightarrow \qquad x_j = 0.$$

This means: Either the primal constraint $x_j \geq 0$ is active (i.e. $x_j = 0$) or the dual constraint $\sum_{i=1}^{m} a_{ij}\lambda_i \leq c_i$ is active (i.e. $\sum_{i=1}^{m} a_{ij}\lambda_i = c_i$). It cannot happen that both constraints are inactive at the same time (i.e. $x_j > 0$ and $\sum_{i=1}^{m} a_{ij}\lambda_i < c_i$).

Proof: The assertion in (i) is an immediate consequence of the weak duality theorem. The assertion in (ii) follows from

$$c^\top x - b^\top\lambda = c^\top x - \lambda^\top b = c^\top x - \lambda^\top A x = \left(c - A^\top\lambda\right)^\top x = \sum_{j=1}^{n} \left(c_j - \sum_{i=1}^{m} a_{ij}\lambda_i \right) x_j.$$

If $c^\top x = b^\top\lambda$, then owing to $x \geq 0$ and $c - A^\top\lambda \geq 0$ every term in the sum has to vanish. If, on the other hand, the complementarity conditions hold, then every term of the sum vanishes and thus $c^\top x = b^\top\lambda$. □

The sufficient optimality condition rises the question, whether $c^\top x = b^\top\lambda$ actually can be obtained. We will answer this question by exploitation of the simplex method. We have seen before that the simplex method with Bland's rule (and assuming $\text{rank}(A) = m$) always terminates, either with an optimal feasible basic solution or with the message that the problem does not have a solution. Let us investigate the first case. Let x be an optimal feasible basic solution of the primal problem (P) with basis index set B, which has been computed by the simplex method using Bland's rule. Hence, it holds $x_B \geq 0$ and $x_N = 0$. Moreover, the optimality criterion

$$\zeta^\top = c_B^\top A_B^{-1} A_N - c_N^\top \leq 0$$

holds. Let λ be defined by

$$\lambda^\top := c_B^\top A_B^{-1},$$

see the revised simplex method. We will show that this λ solves the dual problem! Firstly, it holds

$$c^\top x = c_B^\top x_B = c_B^\top A_B^{-1} b = \lambda^\top b = b^\top \lambda.$$

Secondly, λ is feasible for the dual problem (D) because

$$\begin{aligned} A_B^\top \lambda &= c_B, \\ A_N^\top \lambda - c_N &= A_N^\top \left(A_B^\top\right)^{-1} c_B - c_N = \zeta \leq 0. \end{aligned}$$

The latter is just the primal optimality criterion. Combining both relations yields $A^\top \lambda \leq c$. Hence, we have shown:

> If the primal problem (P) has an optimal solution, then the dual problem (D) has an optimal solution, too, and an optimal dual solution is given by $\lambda^\top = c_B^\top A_B^{-1}$ (there may be other solutions).

Using the dual simplex method, which will be discussed in the next section, the converse can be shown as well. The following theorem is the main result of this section.

Theorem 3.2.4 **(Strong Duality Theorem)**
The primal problem (P) has an optimal solution x if and only if the dual problem (D) has an optimal solution λ. Moreover, the primal and dual objective function values coincide if an optimal solution exists, i.e. $c^\top x = b^\top \lambda$.

Remark 3.2.5
The primal simplex method, compare Algorithm 2.2.7, and the revised simplex method, compare Algorithm 2.5.8, compute vertices x, which are primally feasible, and dual variables λ, which satisfy $c^\top x = b^\top \lambda$ in each step. The primal simplex method stops as soon as λ becomes feasible for the dual problem, i.e. dual feasibility is the optimality criterion for the primal simplex method.

3.3 Dual Simplex Method

The dual simplex method is designed to solve the dual problem

$$(D) \qquad \text{Maximise} \qquad b^\top \lambda \qquad \text{subject to} \qquad A^\top \lambda \leq c$$

of the primal problem (P). Herein, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ are given data and the condition $\text{rank}(A) = m$ is supposed to hold. Of course, by solving (D) we implicitly solve

$$(P) \qquad \text{Minimise} \qquad c^\top x \qquad \text{subject to} \qquad Ax = b, \; x \geq 0$$

as well according to the strong duality theorem.
The dual method is essentially constructed in the same way as the primal simplex method, compare Section 2.2.1 and Algorithm 2.2.7. The main difference is that the dual method computes a sequence of feasible dual basic solutions λ. Such a feasible dual basic solution satisfies

$$A_B^\top \lambda = c_B, \qquad A_N^\top \lambda \leq c_N$$

for a basis index set B and a non-basis index set N. Notice that λ is uniquely determined by the linear equation $A_B^\top \lambda = c_B$ as A_B is supposed to be a basis matrix. Furthermore, notice that the revised simplex method, compare Algorithm 2.5.8, computes the dual variable λ as a by-product (although the λ's in the revised simplex method satisfy the dual constraints solely in the optimal solution and not at intermediate steps). Finally, notice that the condition $A_N^\top \lambda \leq c_N$ is equivalent with the condition

$$\zeta^\top = \lambda^\top A_N - c_N^\top \leq 0,$$

which is nothing else but the optimality criterion used in the primal method.
The dual simplex method even works with the same simplex table as the primal simplex method, but in a transposed sense. Similarly, basis changes are performed such that the following requirements are satisfied:

(i) Dual feasibility:
λ^+ has to remain feasible for the dual problem, i.e.

$$A^\top \lambda \leq c \qquad \Rightarrow \qquad A^\top \lambda^+ \leq c.$$

(ii) Ascent property:
The dual objective function value increases monotonically, i.e.

$$b^\top \lambda^+ \geq b^\top \lambda.$$

Using the same techniques as in Section 2.2.1 one can show that the above requirements are satisfied if the pivot element is chosen as follows:

- Choice of pivot row p: Choose $p \in B$ with $\beta_p < 0$.
- Choice of pivot column q: Choose $q \in N$ such that

$$\frac{\zeta_q}{\gamma_{pq}} = \min \left\{ \frac{\zeta_j}{\gamma_{pj}} \;:\; \gamma_{pj} < 0, \; j \in N \right\}$$

Bland's anti-cycling rule can be applied accordingly.

The following stopping criterions apply:

- Optimality criterion:

  If $\beta_i \geq 0$ holds for all $i \in B$, then the current dual basic solution λ is optimal. The corresponding primal solution is given by $x_B = \beta \geq 0$.

- Unboundedness criterion:

  If there exists $\beta_p < 0$ for some $p \in B$ and $\gamma_{pj} \geq 0$ for all $j \in N$, then the dual problem is unsolvable because the dual objective function is unbounded from above (the feasible set of the primal problem is empty).

- Uniqueness criterion: If in the optimal table $\beta_i > 0$ holds for every $i \in B$, then the dual solution is unique.

We summarise our findings in an algorithm:

Algorithm 3.3.1 (Dual Simplex Method)

(1) Phase 1:

Determine a feasible dual basic solution λ for the dual problem (D) satisfying

$$A_B^\top \lambda = c_B, \qquad \zeta^\top = \lambda^\top A_N - c_N^\top \leq 0,$$

with basis index set B, non-basis index set N, basis matrix A_B, non-basis matrix A_N.

If no feasible dual point exists, STOP. The problem is infeasible.

(2) Phase 2:

(i) Compute $\Gamma = (\gamma_{ij})_{i\in B, j\in N}$, $\beta = (\beta_i)_{i\in B}$, and $\zeta = (\zeta_j)_{j\in N}$ according to

$$\Gamma = A_B^{-1} A_N, \qquad \beta = A_B^{-1} b, \qquad \lambda = \left(A_B^{-1}\right)^\top c_B, \qquad \zeta^\top = \lambda^\top A_N - c_N^\top.$$

(ii) Check for optimality:
If $\beta_i \geq 0$ for every $i \in B$, then STOP. The current feasible dual basic solution λ is optimal (the corresponding primal optimal solution is given by $x_B = \beta$ and $x_N = 0$). The objective function value is $d = b^\top \lambda$.

(iii) Check for unboundedness:
If there exists an index p with $\beta_p < 0$ and $\gamma_{pj} \geq 0$ for every $j \in N$, then the dual problem does not have a solution and the objective function is unbounded from above. STOP.

(iv) *Determine pivot element:*
Choose an index p with $\beta_p < 0$ (according to Bland's rule). p defines the pivot row. Choose an index q (according to Bland's rule) with

$$\frac{\zeta_q}{\gamma_{pq}} = \min\left\{\frac{\zeta_j}{\gamma_{pj}} \; : \; \gamma_{pj} < 0, \; j \in N\right\}.$$

q defines the pivot column.

(v) *Perform basis change:*
Set $B := (B \setminus \{p\}) \cup \{q\}$ and $N := (N \setminus \{q\}) \cup \{p\}$.

(vi) *Go to (i).*

The simplex table in Section 2.3 can be used as well. The update formulae remain valid. The dual method is particularly useful for those problems where a feasible dual basic solution is given, but a feasible primal basic solution is hard to find. For instance, the following problem class is well-suited for the dual simplex method:

$$\text{Minimise} \qquad c^\top x \qquad \text{subject to} \qquad Ax \leq b, \; x \geq 0,$$

where $c \geq 0$ and $b \not\geq 0$. Introduction of a slack variable $y \geq 0$ leads to the standard problem

$$\text{Minimise} \qquad c^\top x \qquad \text{subject to} \qquad Ax + y = b, \; x \geq 0, \; y \geq 0.$$

While $y = b \not\geq 0$ and $x = 0$ is not feasible for the primal problem, the point leads to a feasible dual solution because $\zeta^\top = -c \leq 0$.

Example 3.3.2

Consider the standard LP

$$\textit{Minimise} \qquad c^\top x \qquad \textit{subject to} \qquad Ax = b, \; x \geq 0,$$

with (x_5, x_6, x_7 are slack variables)

$$c = \begin{pmatrix} 5 \\ 3 \\ 3 \\ 6 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad b = \begin{pmatrix} 14 \\ -25 \\ 14 \end{pmatrix}, \quad A = \begin{pmatrix} -6 & 1 & 2 & 4 & 1 & 0 & 0 \\ 3 & -2 & -1 & -5 & 0 & 1 & 0 \\ -2 & 1 & 0 & 2 & 0 & 0 & 1 \end{pmatrix}.$$

An initial feasible dual basic solution is given by the index set $B = \{5, 6, 7\}$. The dual simplex method yields:

Initial table:

	x_1	x_2	x_3	x_4	
x_5	-6	1	2	4	14
x_6	3	-2	-1	$\boxed{-5}$	-25
x_7	-2	1	0	2	14
	-5	-3	-3	-6	0

Table 1:

	x_1	x_2	x_3	x_6	
x_5	-3.6	$\boxed{-0.6}$	1.2	0.8	-6
x_4	-0.6	0.4	0.2	-0.2	5
x_7	-0.8	0.2	-0.4	0.4	4
	-8.6	-0.6	-1.8	-1.2	30

Table 2:

	x_1	x_5	x_3	x_6	
x_2	6	$-5/3$	-2	$-4/3$	10
x_4	-3	$2/3$	1	$1/3$	1
x_7	-2	$1/3$	0	$2/3$	2
	-5	-1	-3	-2	36

This table is optimal as $x = (0, 10, 0, 1, 0, 0, 2)^\top$ is feasible for the primal problem and the optimal objective function value is 36. A corresponding dual solution is obtained by solving the linear equation $A_B^\top \lambda = c_B$. We obtain $\lambda = (-1, -2, 0)^\top$.

Remark 3.3.3

The dual simplex method, compare Algorithm 3.3.1 computes vertices λ, which are feasible for the dual problem, and primal variables x, which satisfy $c^\top x = b^\top \lambda$ in each step. The dual simplex method stops as soon as x becomes feasible for the primal problem, i.e. primal feasibility is the optimality criterion for the dual simplex method.

3.4 Sensitivities and Shadow Prices

The solutions of the dual problem possess an important interpretation for economical applications. Under suitable assumptions they provide the sensitivity of the primal objective function value $c^\top x$ with respect to perturbations in the vector b. In economy, the dual solutions are known as shadow prices.

In practical applications, the vector b often is used to model capacities (budget, resources, size, etc.) while the objective function often denotes the costs. An important economical question is the following: How does a variation in b (i.e. a variation in budget, resources, size) influence the optimal objective function value (i.e. the costs)?

To answer this question, the primal problem (P) is embedded into a family of perturbed linear programs:

Definition 3.4.1 (Perturbed problem (P_δ))
For an arbitrary $\delta \in \mathbb{R}^m$ the LP

$$(P_\delta) \qquad \text{Minimise} \quad c^\top x \quad \text{subject to} \quad Ax = b + \delta,\ x \geq 0$$

with $A \in \mathbb{R}^{m\times n}$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ is called perturbed primal problem. The set

$$M_\delta := \{x \in \mathbb{R}^n \mid Ax = b + \delta,\ x \geq 0\}$$

is the feasible set of the perturbed primal problem.

Obviously, (P_δ) with $\delta = 0$ is identical with the primal problem (P) which is referred to as the unperturbed problem. It is clear that the solution of the perturbed problem depends somehow on δ, i.e. $x = x(\delta)$. We are now interested in the following question: How does the optimal solution $x(\delta)$ of the perturbed problem depend on δ for δ sufficiently close to zero?

Let us approach the problem graphically first.

Example 3.4.2 (compare Examples 1.1.1, 1.3.1)
A farmer intends to plant 40 acres with sugar beets and wheat. He can use up to 312 working days to achieve this. For each acre his cultivation costs amount to 40 pounds for sugar beets and to 120 pounds for wheat. For sugar beets he needs 6 working days per acre and for wheat 12 working days per acre. The profit amounts to 100 pounds per acre for sugar beets and to 250 pounds per acre for wheat. As the farmer wants to consider unforeseen expenses in his calculation, he assumes that he has a maximal budget of $2400 + \delta$ pounds, where $\delta \in \mathbb{R}$ is a perturbation potentially caused by unforeseen expenses. Of course, the farmer wants to maximise his profit (resp. minimise the negative profit).

The resulting canonical linear program reads as follows (compare Example 1.3.1):

Minimise $f(x_1, x_2) = -100x_1 - 250x_2$ subject to the constraints

$$x_1 + x_2 \leq 40, \quad 40x_1 + 120x_2 \leq 2400 + \delta, \quad 6x_1 + 12x_2 \leq 312, \quad x_1, x_2 \geq 0.$$

In Example 1.3.1 we solved the unperturbed problem for $\delta = 0$ graphically:

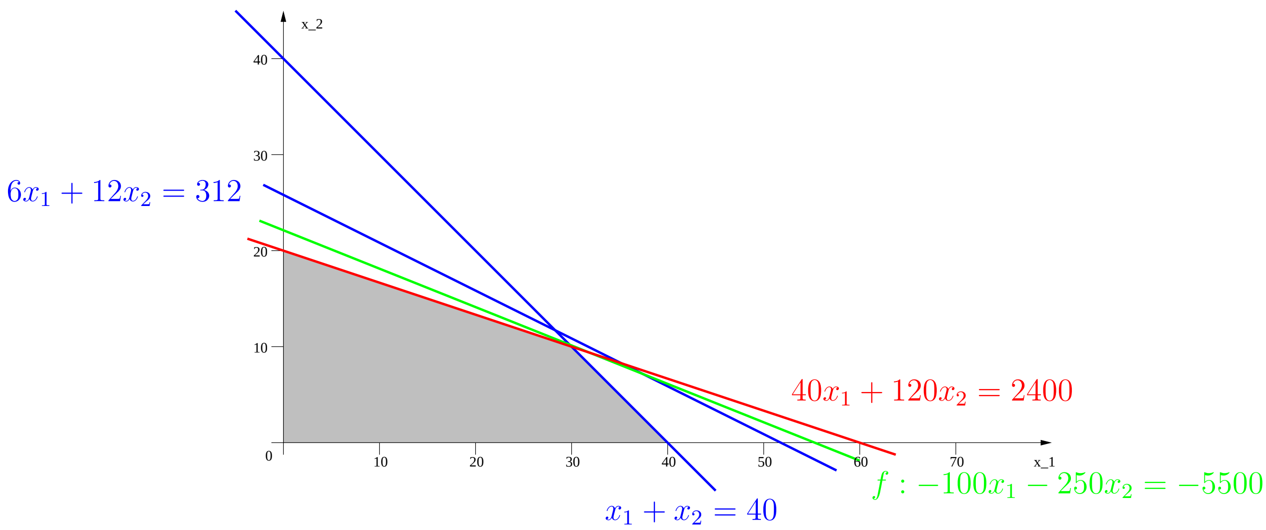


Figure 3.1: Solution of the unperturbed problem for $\delta = 0$. The optimal solution is $(x_1, x_2)^\top = (30, 10)^\top$ with objective function value -5500.

The optimal solution is obtained at the point where the first and second constraint intersect (this corresponds to x_1, x_2 being basic variables in the simplex method). What happens if perturbations $\delta \neq 0$ are considered? Well, δ influences only the second constraint. Moreover, the slope of this constraint is not affected by changing δ. Hence, changing δ results in lines which are in parallel to the red line in Figure 3.1. Figure 3.2 depicts the situation for $\delta = -600$.

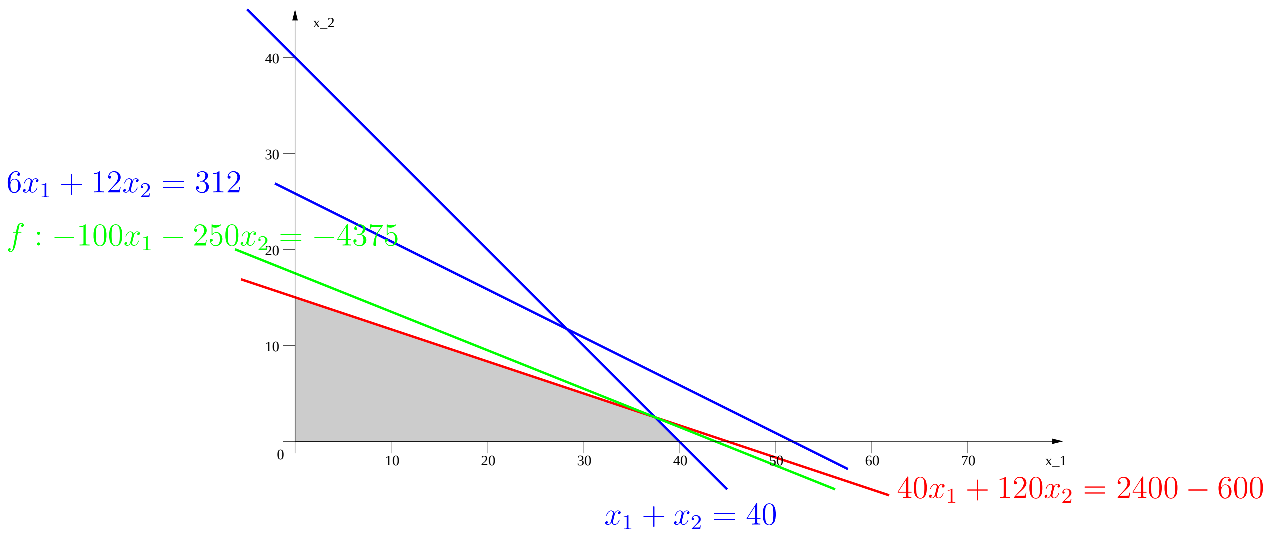


Figure 3.2: Solution of the problem for $\delta = -600$. The optimal solution is $(x_1, x_2)^\top = (37.5, 2.5)^\top$ with objective function value -4375.

Clearly, the feasible set and the optimal solution have changed. But still, the optimal solution is obtained at the point where the first and second constraint intersect (again, x_1, x_2 are among the basic variables in the simplex method).

What happens if we further reduce δ, say $\delta = -1200$? Figure 3.3 depicts the situation for $\delta = -1200$.

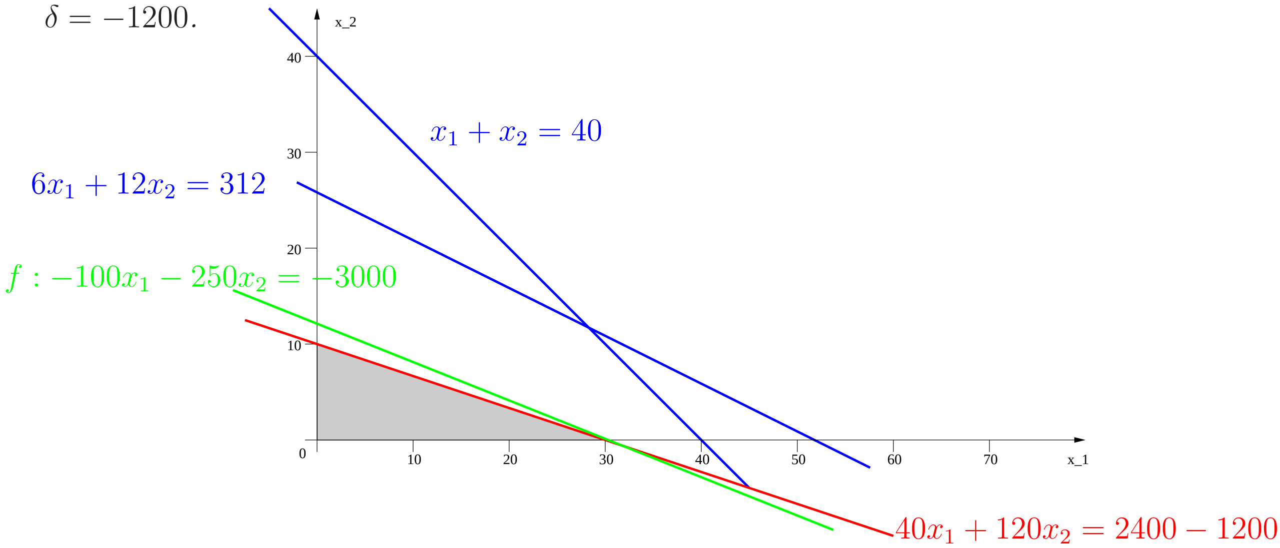


Figure 3.3: Solution of the problem for $\delta = -1200$. The optimal solution is $(x_1, x_2)^\top = (30, 0)^\top$ with objective function value -3000.

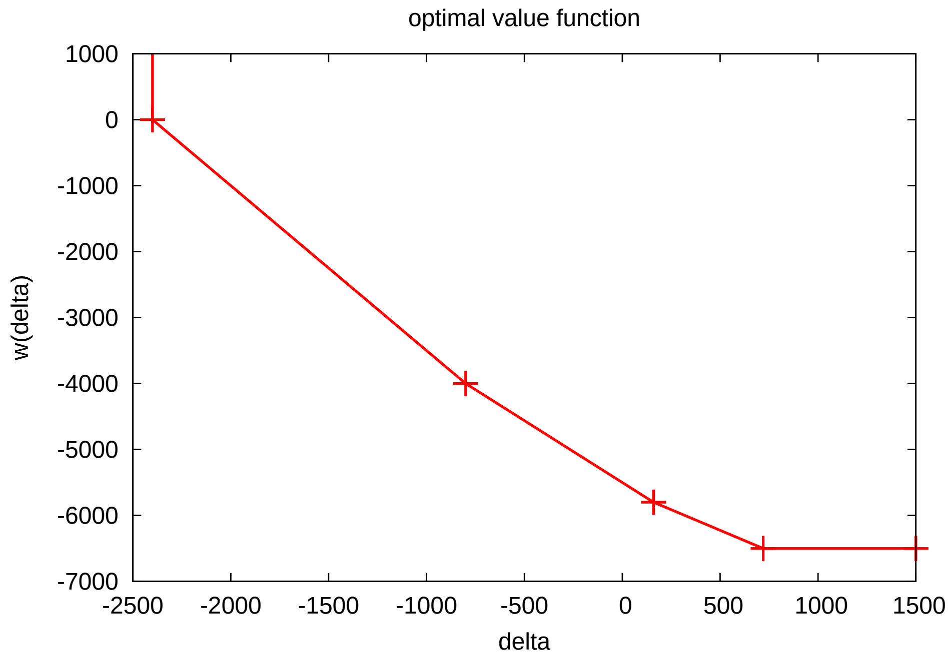
Clearly, the feasible set and the optimal solution have changed again. But now, the optimal solution is obtained at the point where the second constraint intersects with the constraint $x_2 = 0$. So, the structure of the solution has changed (now, x_2 became a non-basic variable and thus a switch of the basis index set occurred).
Finally, we could ask ourselves, how the optimal objective function value depends on δ? A graphical investigation of the problem for all values of δ yields the optimal solutions

$$\begin{pmatrix} x_1(\delta) \\ x_2(\delta) \end{pmatrix} = \begin{cases} \begin{pmatrix} 0 \\ 26 \end{pmatrix}, & \text{if } 720 \le \delta, \\ \begin{pmatrix} 36 - \frac{1}{20}\delta \\ 8 + \frac{1}{40}\delta \end{pmatrix}, & \text{if } 160 \le \delta < 720, \\ \begin{pmatrix} 30 - \frac{1}{80}\delta \\ 10 + \frac{1}{80}\delta \end{pmatrix}, & \text{if } -800 \le \delta < 160, \\ \begin{pmatrix} 60 + \frac{1}{40}\delta \\ 0 \end{pmatrix}, & \text{if } -2400 \le \delta < -800, \\ \text{no solution}, & \text{if } \delta < -2400, \end{cases}$$

and the optimal value function, cf. Definition 3.4.3,

$$w(\delta) = c^\top x(\delta) = \begin{cases} -6500, & \text{if } 720 \le \delta, \\ -5600 - 1.25\delta, & \text{if } 160 \le \delta < 720, \\ -5500 - 1.875\delta, & \text{if } -800 \le \delta < 160, \\ -6000 - 2.5\delta, & \text{if } -2400 \le \delta < -800, \\ \infty, & \text{if } \delta < -2400, \end{cases}$$

which is piecewise linear and convex:



In the sequel we restrict the discussion to perturbations of the vector b only. Perturbations in A and c can be investigated as well, but the situation is more complicated. We consider the perturbed primal problem in Definition 3.4.1, i.e.

$$\text{Minimise} \quad c^\top x \quad \text{subject to} \quad Ax = b + \delta,\ x \geq 0.$$

Obviously, for $\delta = 0$ the primal problem (cf. Definition 3.1.2) arises. This problem is referred to as the unperturbed problem or nominal problem.

Each perturbation δ may be assigned the corresponding optimal solution (if it exists), i.e.

$$\delta \mapsto x(\delta).$$

Introducing this function $x(\delta)$ into the objective function $c^\top x$ leads to the optimal value function $w(\delta)$, which is defined as follows.

Definition 3.4.3 (Optimal value function)

The optimal value function is defined as

$$w(\delta) := \begin{cases} \inf\{c^\top x \mid Ax = b + \delta,\ x \geq 0\}, & \text{if } M_\delta \neq \emptyset, \\ \infty, & \text{if } M_\delta = \emptyset. \end{cases}$$

We now intend to investigate the optimal solution $x(\delta)$ of the perturbed problem 3.4.1 in a small neighbourhood of the unperturbed solution at $\delta = 0$, i.e. we consider perturbations $\delta \in \mathbb{R}^m$ sufficiently close to zero.

Moreover, we will assume the following:

- the unperturbed problem with $\delta = 0$ possesses the feasible basic solution $x_B(0) = A_B^{-1}b$, $x_N(0) = 0$ with basis index set B and non-basis index set N.
- the solution is not degenerate, i.e. $x_B(0) > 0$.

An extension to degenerate solutions is possible, but more involved.
Now we introduce perturbations $\delta \neq 0$ sufficiently close to zero and define

$$x_B(\delta) := A_B^{-1}(b + \delta), \qquad x_N(\delta) := 0. \tag{3.1}$$

The point given by $x_B(\delta)$ and $x_N(\delta)$ is feasible as long as

$$x_B(\delta) = A_B^{-1}(b + \delta) = x_B(0) + A_B^{-1}\delta \geq 0$$

holds. As x is supposed to be non-degenerate, i.e. $x_B(0) = A_B^{-1}b > 0$, feasibility can be guaranteed for perturbations δ sufficiently close to zero owing to the continuity of the function $\delta \mapsto x_B(0) + A_B^{-1}\delta$.
What about the optimality criterion for $x_B(\delta)$ and $x_N(\delta)$? Interestingly, perturbations δ in b do not influence the optimality criterion of the simplex method at all because the vector ζ does not depend on b:

$$\zeta^\top = c_B^\top A_B^{-1} A_N - c_N^\top.$$

As $x_B(0)$ was supposed to be optimal it holds $\zeta \leq 0$. This optimality condition is still satisfied for $x_B(\delta)$ and $x_N(\delta)$ in (3.1) and hence, (3.1) is optimal for the perturbed problem! The corresponding optimal objective function value computes to

$$w(\delta) = c_B^\top x_B(\delta) = c_B^\top x_B(0) + c_B^\top A_B^{-1}\delta = w(0) + \lambda^\top \delta,$$

where $\lambda = (A_B^{-1})^\top c_B$ is a solution of the dual problem. Hence:

Theorem 3.4.4 (Sensitivity Theorem)
Let rank$(A) = m$ and let the unperturbed problem possess an optimal feasible basic solution with basis index set B and non-basis index set N which is not degenerate. Then, for all perturbations $\delta = (\delta_1, \ldots, \delta_m)^\top$ sufficiently close to zero,

$$x_B(\delta) = A_B^{-1}(b + \delta), \qquad x_N(\delta) = 0,$$

is an optimal feasible basic solution for the perturbed problem and it holds

$$\Delta w := w(\delta) - w(0) = \lambda^\top \delta = \sum_{i=1}^{m} \lambda_i \delta_i.$$

The dual solution $\lambda^\top = c_B^\top A_B^{-1}$ indicates the sensitivity of the optimal objective function value w.r.t. perturbations in b.

The sensitivity theorem shows:

- If $|\lambda_i|$ is large, then perturbations δ_i with $|\delta_i|$ small have a comparatively large influence on the optimal objective function value. The problem is sensitive w.r.t. to perturbations.
- If $|\lambda_i|$ is small, then perturbations δ_i with $|\delta_i|$ small have a comparatively small influence on the optimal objective function value. The problem is not sensitive w.r.t. to perturbations.
- If $\lambda_i = 0$, then perturbations δ_i close to zero have no influence on the optimal objective function value.

The sensitivity theorem states in particular, that under the assumptions of the theorem, $x_B(\delta)$ and $w(\delta)$ are continuously differentiable at $\delta = 0$. Differentiation of the optimal value function w.r.t. δ yields

Shadow price formula:

$$w'(\delta) = \lambda.$$

Example 3.4.5 **(compare Example 3.4.2)**

We consider again Example 3.4.2:

Minimise $f(x_1, x_2) = -100x_1 - 250x_2$ *subject to the constraints*

$$x_1 + x_2 \le 40, \quad 40x_1 + 120x_2 \le 2400 + \delta, \quad 6x_1 + 12x_2 \le 312, \quad x_1, x_2 \ge 0.$$

The simplex method for the unperturbed problem with $\delta = 0$ *yields:*

Initial table:

	x_1	x_2	
x_3	$\boxed{1}$	1	40
x_4	40	120	2400
x_5	6	12	312
	100	250	0

Table 1:

	x_3	x_2	
x_1	1	1	40
x_4	−40	$\boxed{80}$	800
x_5	−6	6	72
	−100	150	−4000

Table 2:

	x_3	x_4	
x_1	1.5	−0.0125	30
x_2	−0.5	0.0125	10
x_5	−3	−0.075	12
	−25	−1.875	−5500

Table 2 is optimal. The optimal solution $x = (30, 10, 0, 0, 12)^\top$ *is not degenerate. The corresponding dual solution is*

$$\lambda^\top = c_B^\top A_B^{-1} = (-25, -1.875, 0),$$

where $B = \{1, 2, 5\}$ *and*

$$c_B = \begin{pmatrix} -100 \\ -250 \\ 0 \end{pmatrix}, \qquad A_B = \begin{pmatrix} 1 & 1 & 0 \\ 40 & 120 & 0 \\ 6 & 12 & 1 \end{pmatrix}, \qquad A_B^{-1} = \begin{pmatrix} \frac{3}{2} & -\frac{1}{80} & 0 \\ -\frac{1}{2} & \frac{1}{80} & 0 \\ -3 & -\frac{3}{40} & 1 \end{pmatrix}.$$

The sensitivity theorem states:

(i)

$$\underbrace{\begin{pmatrix} x_1(\delta) \\ x_2(\delta) \\ x_5(\delta) \end{pmatrix}}_{=x_B(\delta)} = \underbrace{\begin{pmatrix} x_1(0) \\ x_2(0) \\ x_5(0) \end{pmatrix}}_{=x_B(0)} + A_B^{-1} \begin{pmatrix} 0 \\ \delta \\ 0 \end{pmatrix} = \begin{pmatrix} 30 \\ 10 \\ 12 \end{pmatrix} + \delta \begin{pmatrix} -\frac{1}{80} \\ \frac{1}{80} \\ -\frac{3}{40} \end{pmatrix}$$

is optimal for δ *sufficiently close to zero. How large may the perturbation* δ *be? Well,* $x_B(\delta)$ *has to be feasible, i.e.*

$$30 - \frac{1}{80}\delta \geq 0, \quad 10 + \frac{1}{80}\delta \geq 0, \quad 12 - \frac{3}{40}\delta \geq 0.$$

These three inequalities are satisfied if $\delta \in [-800, 160]$. *Hence, for all* $\delta \in [-800, 160]$, *the above* $x_B(\delta)$ *is optimal.*

(ii)

$$\Delta w = w(\delta) - w(0) = \lambda^\top \begin{pmatrix} 0 \\ \delta \\ 0 \end{pmatrix} = -1.875\delta.$$

The change of the negative profit w.r.t. δ *amounts to* -1.875δ.

Compare this with our graphical investigation in Example 3.4.2!

Remark 3.4.6 (Caution!)

The assumption that the unperturbed solution is not degenerate is essential and cannot be *dropped! The above analysis only works, if the index sets B and N don't change! This can only be guaranteed for perturbations δ sufficiently close to zero. For large perturbations or in the degenerate case, the index sets B and N usually do change and the optimal value function is not differentiable at those points, where the index sets change. Example 3.4.2 clearly shows, that this does happen.*

Bibliography

[Bla77] Bland, R. G. *New finite pivoting rules for the simplex method*. Mathematics of Operations Research, 2 (2); 103–107, 1977.

[Bor87] Borgwardt, K. H. *The simplex method – A probabilistic analysis*. Springer, Berlin-Heidelberg-New York, 1987.

[Chv83] Chvatal, V. *Linear Programming*. W. H. Freeman and Comp., New York, 1983.

[Dan98] Dantzig, G. B. *Linear Programming and Extensions*. Princeton University Press, new ed edition, 1998.

[Gas03] Gass, S. *Linear Programming – Methods and Applications*. Dover Publications, 5th edition, 2003.

[Hof53] Hoffman, A. J. *Cycling in the simplex algorithm*. National Bureau of Standards Report, 2974, 1953.

[Kar84] Karmarkar, N. *A new polynomial-time algorithm for linear programming*. AT&T Bell Laboratories, Murray Hill, New Jersey 07974, 1984.

[Kha79] Khachiyan, L. G. *A polynomial algorithm in linear programming*. Doklady Akademiia Nauk SSSR, 244; 1093–1096, 1979.

[KM72] Klee, V. and Minty, G. J. *How good is the simplex algorithm?*. In *Inequalities III* (O. Shisha, editor), pp. 159–175. Academic Press, New York, 1972.

[MS69] Marshall, K. T. and Suurballe, J. W. *A note on cycling in the simplex method*. Naval Res. Logist. Quart., 16; 121–137, 1969.

[NW99] Nocedal, J. and Wright, S. J. *Numerical optimization*. Springer Series in Operations Research, New York, 1999.

[PS98] Papadimitriou, C. H. and Steiglitz, K. *Combinatorial Optimization–Algorithms and Complexity*. Dover Publications, 1998.

[Tod02] Todd, M. J. *The many facets of linear programming*. Mathematical Programming, Series B, 91; 417–436, 2002.

[Win04] Winston, W. L. *Operations Research: Applications and Algorithms*. Brooks/Cole–Thomson Learning, Belmont, 4th edition, 2004.

[Wri97] Wright, S. E. *Primal-Dual Interior-Point Methods*. SIAM, Philadelphia, 1997.